Less Machine (=) More Vision

Approaches towards Practical and Efficient Machine Vision with Applications in Face Analysis

Gudi, A.A.

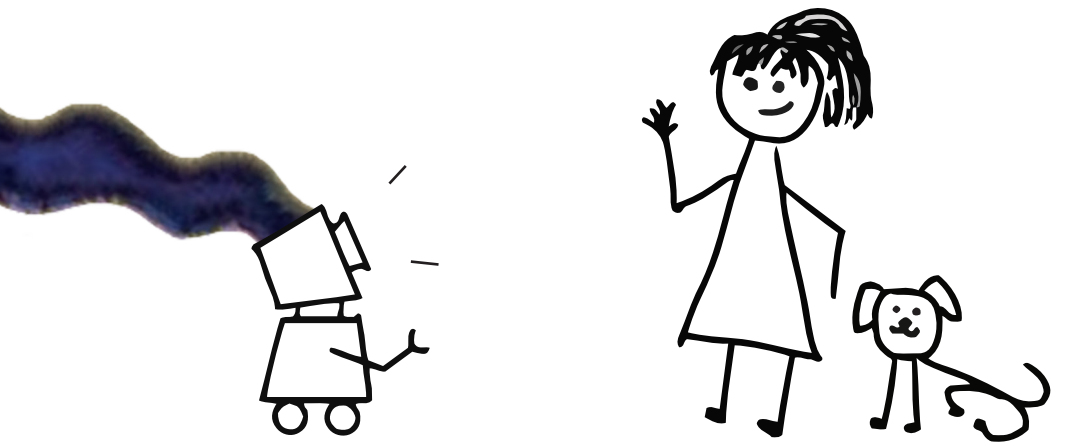
DOI
10.4233/uuid:8dbbf209-ef24-48c4-bfe2-9b029e2f97dc

Publication date
2022

Document Version
Final published version

Citation (APA)
Gudi, A. A. (2022). *Less Machine (=) More Vision: Approaches towards Practical and Efficient Machine Vision with Applications in Face Analysis*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:8dbbf209-ef24-48c4-bfe2-9b029e2f97dc

less machine (=) more vision



AMOGH GUDI

Less Machine (=) More Vision

Approaches towards
Practical and Efficient Machine Vision with Applications in Face Analysis

Less Machine (=) More Vision

Approaches towards Practical and Efficient Machine Vision with Applications in Face Analysis

Dissertation

for attainment of the degree of doctor
at the Delft University of Technology,
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Monday, 3rd October 2022, at 3:00 pm

by

Amogh Anirudh GUDI

Master of Science in Artificial Intelligence,
University of Amsterdam, The Netherlands,
born in Udhampur, India.

This dissertation has been approved by the promotors.

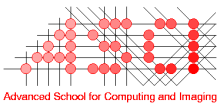
Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. ir. M.J.T. Reinders,	Delft University of Technology, promotor
Dr. J.C. van Gemert,	Delft University of Technology, copromotor

Independant members:

Prof. dr. P.H.N. de With,	Eindhoven Institute of Technology, The Netherlands
Prof. dr. D.M. Gavrila,	Delft University of Technology, The Netherlands
Dr. F.M. Vos,	Delft University of Technology, The Netherlands
Dr. ir. R.W. Poppe,	Utrecht University, The Netherlands
Dr. H. Dibeklioğlu,	Bilkent University, Turkey
Prof. dr. C.M. Jonker,	Delft University of Technology, reserve member

The work in this thesis has been funded by Vicarious Perception Technologies (VicarVision).



This work was carried out in the Advanced School for Computing and Imaging (ASCI).
ASCI dissertation series number 439.

Printed by:	ProefschriftMaken
Cover:	Artwork based on *All universe in your head* by 'Rain747' on Artmonía.
Style:	TU Delft House Style, with modifications.

ISBN 978-94-6366-602-2

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

Terugkomen is niet hetzelfde als blijven. | *"Coming back is not the same as staying."*

Belle van Zuylen

Seen in the city of Amsterdam between *Het Noord* and *Singel.* Artwork by Regina Verhagen.

Contents

Contributing Publications

Chapter 2: Weakly Supervised Single-Shot Localization

Gudi, A., Van Rosmalen, N., Loog, M., & Van Gemert, J. (2017). Object extent pooling for weakly supervised single-shot localization. In British Machine Vision Conference 2017, BMVC 2017 (British Machine Vision Conference 2017, BMVC 2017). BMVA Press.
https://doi.org/10.5244/c.31.36

Chapter 3: Input & Data Efficiency in Webcam Gaze Tracking

Gudi, A., li, X., & van Gemert, J. (2020). Efficiency in Real-time Webcam Gaze Tracking. In A. Bartoli, & A. Fusiello (Eds.), Computer Vision – ECCV 2020 Workshops: Proceedings (1 ed., pp. 529 - 543). (Part of the Lecture Notes in Computer Science book series (LNCS, volume 12535) Also part of the Image Processing, Computer Vision, Pattern Recognition, and Graphics book sub series (LNIP, volume 12535); Vol. 12535). Springer.
https://doi.org/10.1007/978-3-030-66415-2_34

Chapter 4: Exploiting Spatial Context for Anomaly Detection & Feat. Learning

Gudi, A., Büttner, F., & van Gemert, J. (2019). Proximally Sensitive Error for Anomaly Detection and Feature Learning. Extended abstract, ICT.OPEN, Hilversum, 2019. ArXiv:2206.00506.
https://arxiv.org/abs/2206.00506

Chapter 5: Prior Knowledge driven Efficient Vital Signs Estimation from Faces

Gudi, A., Bittner, M., & van Gemert, J. (2020). Real-Time Webcam Heart-Rate and Variability Estimation with Clean Ground Truth for Evaluation. Applied Sciences, 10(23), 1-24. [8630].
https://doi.org/10.3390/app10238630

Gudi, A., Bittner, M., Lochmans, R., & van Gemert, J. (2019). Efficient real-time camera based estimation of heart rate and its variability. In Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019 (pp. 1570-1579). [9022193].
https://doi.org/10.1109/ICCVW.2019.00196

Summary

Machines that interact with humans can do so better if they can also visually understand us, but they have limited resources to do so. The main topic of this dissertation is contrasting the use of resources by machine vision systems against the accuracy obtained by them. This thesis focuses on reducing the need for data, memory, and computation in real-world machine vision systems, applied to human observation and face analysis.

This dissertation tackles annotation effort by exploring how weakly-supervised object /person detectors can be improved. Findings show that prior knowledge about objects' bounds in images helps the detector learn the spatial extent of objects using only weak image-level labels. The proposed implementation enables single-shot detection, thus improving computational efficiency of this data-efficient method.

The thesis also demonstrates how prior knowledge about eye locations can be used to reduce the computational burden of gaze tracking: non-vital parts of the input image can be discarded without losing accuracy. Additionally, this thesis finds how *a priori* known geometrical relations can be leveraged upon to project gaze onto a screen with little human annotation effort.

Findings of this dissertation further suggest that spatial structures in images can be exploited for improving efficiency of vision tasks. The proposed solution allows for learning detection of facial occlusions and anomalies from only a few examples. Results also indicate that this solution can be used as a loss function for unsupervised pre-training of neural networks when resources are constrained.

Lastly, this thesis showcases how prior know-how about blood-flow physiology in faces can be applied in a camera-based vital signs estimator. Even when data is available, this hand-crafted method performs better than deep learning methods — both in terms of accuracy and efficiency. At the same time, the results also reveal the pitfalls of assumptions made in the prior knowledge when exposed to more complex tasks — such as video compression noise filtering.

Through its common theme of incorporating prior knowledge, this dissertation brings attention to the costs incurred by machine vision systems to achieve high accuracy.

Samenvatting

Machines kunnen beter met mensen samenwerken als ze de wereld kunnen begrijpen met hun ogen. Een bottleneck voor dit begrip wordt gevormd door de alle middelen die zelflerende systemen hiervoor nodig hebben. In dit proefschrift bestuderen we de accuraatheid van machines versus de middelen die deze systemen nodig hebben. We richten ons op het verminderen van deze bronnen, zoals de hoeveelheid data, geheugen en rekenkracht van machines. We bestuderen systemen die echt in de wereld worden toegepast, voor gezichtsuitdrukking-analyse en menselijke observatie.

In dit proefschrift kijken we hoe we met minder annotaties nog steeds goede systemen kunnen bouwen, door simpelere annotaties van beelden te gebruiken. Simpele annotaties zijn bijvoorbeeld slechts annotaties waar alleen wordt geconstateerd dat er een voorwerp in een beeld zichtbaar is, maar niet waar het voorwerp zich bevind. Verder voegen we voorkennis toe: namelijk waar objecten zich vaak bevinden in beelden. Met deze twee elementen kan een zelflerend systeem alsnog uit zich zelf leren hoe groot voorwerpen zijn. Ons gebouwde systeem kan door slechts één keer te kijken naar een beeld ("single shot") alsnog meerdere objecten tegelijkertijd detecteren, waardoor de rekentijd en hoeveelheid data beiden worden verminderd van deze zelflerende systemen.

We gebruiken ook dezelfde soort voorkennis, namelijk waar ogen zich bevinden in beelden, om de hoeveelheid rekentijd voor het volgen van de blik van mensen te verminderen. Zo kan ons systeem een deel van het beeld weggooien om herkenning te versnellen zonder dat de accuraatheid hier onder lijdt. Verder kan ook voorkennis van de geometrie tussen de mens en een scherm worden gebruikt om de blik te projecteren op het scherm door een systeem dat slechts weinig data nodig heeft om te leren.

De resultaten van dit proefschrift suggereren dat de structuur in beelden kan worden gebruikt om kijkende systemen sneller te maken. Zo ontwerpen we een systeem dat kan herkennen of een gezicht zichtbaar is of niet, of deels wordt bedekt, en kan ook rariteiten in gezichtsafbeeldingen herkennen. Dit is zelfs mogelijk als er weinig data beschikbaar is. Onze wiskundige functie die wordt gebruikt om te leren kan ook worden gebruikt voor het pre-trainen van neurale netwerken in dat geval.

Tenslotte gebruiken we ook voorkennis van de biologie, namelijk hoe bloed zich gedraagt in gezichten, en gebruiken dit om een nieuw systeem te bouwen dat met een camera de hartslag kan meten. Zelfs wanneer er veel data beschikbaar is, is ons systeem dat voorkennis gebruikt, beter dan een diep neuraal netwerk. Ons systeem is niet alleen accurater maar kan zelfs nog goed leren als er weinig data beschikbaar is. Tegelijkertijd illustreren onze resultaten ook dat voorkennis ook een probleem kan vormen, vooral als de voorkennis niet meer geldig is een nieuwe complexere situaties. Dit vormde vooral een probleem bij video compressie en het bouwen van een ruisfilter.

De hoofdstukken in dit proefschrift draaien om het gebruik van voorkennis voor het maken van zelflerende systemen. Verder benadrukken we de enorme kosten die nodig zijn om accurate zelflerende systemen te bouwen.

सारांश

इंसानों के साथ परस्पर प्रभाव डालने वाले यंत्र यह कार्य बेहतर कर सकते है अगर वह हमें दृष्टिगत रूप से भी समझ सके, परन्तु उनके पास यह करने के लिए सीमित साधन हैं। इस शोध लेख का मुख्य विषय यंत्र दृष्टि प्रणालियों (*"मशीन विज़न सिस्टम्स"*) द्वारा साधनों के उपयोग को उनके द्वारा प्राप्त सटीकता के साथ तुलना करना है। यह लेख वास्तविक दुनिया के यंत्र दृष्टि प्रणालियों में डाटा/जानकारी, मेमोरी/स्मृति, और अभिगणना की मांग को घटाने पर केंद्रित है, जो मनुष्यों के निरीक्षण और चेहरों की विश्लेषण के लिए लागू किए जाते हैं।

यह शोध लेख अंकन कार्य से यह खोज कर के निपटता है कि किस तरह कमजोर–पर्यवेक्षित वास्तु / व्यक्ति संसूचकों को बेहतर बनाया जा सके। जाँच–परिणाम यह दिखाता है कि चित्रों में वस्तुओं के सीमाओं का पूर्व ज्ञान केवल कमजोर छवि–स्तर के अंकितकों का उपयोग कर के वस्तुओं की स्थानिक विस्तार सीखने में संसूचक की मदद करता है। प्रस्तावित कार्यान्वयन चित्र को एक ही बार देख के (*"सिंगल–शॉट"*) आकलन संभव बनता है, जिस द्वारा इस डाटा–दक्ष विधि की अभिगणनीय दक्षता सुधरती हैं।

यह लेख यह भी प्रदर्शित करता है कि कैसे आँखों के स्थान का पूर्व ज्ञान का प्रयोग निगाहों के अनुमान लगाने के कार्य में अभिगणनीय बोझ को घटाने के लिए किया जा सकता है — इनपुट छवि के गैर–महत्वपूर्ण भागों को बिना सटीकता खोये निकाला जा सकता है। साथ ही, यह शोध पता लगता है कि कैसे नज़रों को एक चित्रपट पर प्रक्षेपित करने में पूर्व ज्ञात ज्यामितीय संबंधों का लाभ उठाया जा सकता है।

इस शोध लेख के निष्कर्ष आगे यह सुझाव देते हैं कि यंत्र दृष्टि सम्बन्धी कार्यों की दक्षता बढ़ने के लिए चित्रों में स्थानिक संरचनाओं का फायदा उठाया जा सकता है। प्रस्तावित समाधान केवल कुछ ही उदाहरणों से संसूचकता सीखना मुमकिन बनाता है। परिणाम यह भी दर्शाते हैं कि यह समाधान तंत्रिका जालो (*"न्यूरल नेटवर्क्स"*) की गैर–पर्यवेक्षित पूर्व–प्रशिक्षण (*"अंसूपर्वाइज़ड प्री–ट्रेनिंग"*) में एक *"लॉस"* फलन के तौर पर इस्तेमाल किया जा सकता है जब संसाधनों की विवशता हो।

अंततः, यह लेख यह दर्शाता है कि कैसे चेहरों में रक्त प्रवाह के बारे में पूर्व जानकारी को एक कैमरा–आधारित जीवन–संकेत आगणक में लागू किया जा सकता हैं। डाटा उपलब्ध होने पर भी, यह हस्तनिर्मित विधि *"डीप–लर्निंग"* विधियों से बेहतर प्रदर्शन देता हैं — सटीकता और दक्षता दोनों के मामले में। साथ ही, परिणाम ज्यादा जटिल कार्यों के सामने पूर्व ज्ञान में छुपी धारणाओं के नुकसानों का भी खुलासा करते हैं — जैसे कि वीडियो कम्प्रेशन शोर की छनाई।

पूर्व ज्ञान के समावेश के सामान्य विषय के माध्यम से, यह शोध लेख यंत्र दृष्टि प्रणालियों द्वारा उच्च सटीकता प्राप्त करने के लिए किए गए लागत पर ध्यान आकर्षित करता है।

Acknowledgments

Welcome to the Amogh Gudi Ph.D. Awards, the only award show honouring the most amazing and exceptional people in the professional and social bubbles of Dr. Amogh A Gudi (strongly assuming they decided to grant him the degree). Amogh Gudi had a great time on his Ph.D. journey (despite of his exaggerated complaints), and this wouldn't have been the case without the wonderful humans around him. This award ceremony intends to honour these people.

Before beginning with the esteemed awards, a special message from Amogh:
"First and foremost, I'd like to remember and thank the late Marten den Uyl, founder of SMR & VicarVision, for believing in me and for enabling and encouraging me to pursue this doctorate. Marten — your enthusiasm inspired me towards research and you have played a pivotal role in shaping my career. I will always be grateful for this."

With this, let's kick off the round of award!

In 2016, it was very nice of the supervisor Jan van Gemert to adopt Amogh as a PhD student when he hardly knew him (Amogh followed Jan's CV course during his masters at UvA, but never attended a single class). "Funny-stuff apart, Jan is an brilliant supervisor and working with him is a constant learning experience", says Amogh. This awards him the ***Favourite Supervisor Award***, congratulations Jan! Speaking of brilliant educational leadership, Marcel Reinders gets the ***Nicest Professor Award***; Marcel provided excellent guidance, encouragement and care to Amogh Gudi as his promotor and the head of PRB. Amogh adds, "I somehow dodged the infamous PhD burnout, and my promotors are partly to thank for it." Apart from his direct advisors, Amogh also had the pleasure of interacting with some other excellent educators at TU Delft - Marco Loog and David Tax. Amogh learnt a lot of actual machine learning from them (not what these deep learners do, as Marco would say). This gets them the prestigious ***Amazing Educators Award***.

The next award is for a time when Amogh had just started his PhD journey in the scary corridors of the EWI building. This is when the kind PhD students and staff of PRB — including Alex, Alex, Ekin, Laura, Wenjie, Silvia, Wouter, Christian, and several others — left no stone unturned to welcome him, and for this they are all given the ***Warmest Welcomers Award***. Amogh was of course not alone on this PhD journey, there were several co-travellers with him at PRB. Some started together with Amogh — Tom, Christine, Stavros — while some he met later along the journey — Osman, Ziqi, Yancong, Yunqiang, Yeshwanth, Chirag, Nikolaas, Marian, Robert-Jan, Atilla, Xin, Ombretta, Xiangwei, Jin, Stephanie, Jose, Arman, Rickard, Burak, and Bernd. Several ideas in Amogh's PhD have roots in the discussions and conversations with these fellow PhD'ers. They all deserve the ***Cleverest Crew Award*** for the being there for each other and for Amogh. Special

recognition goes to Tom, Nikolaas, Marian and Ziqi for voluntarily spending time with Amogh even outside of work. One cannot forget the wonderful staff members at PRB, who made this PhD journey easier - Silvia, Nergis, Xucong, Hadi, Seyran, Hayley, and Jesse. They gave Amogh all the advice and help he needed, for which they are awarded the ***Wisest Advisors Award***. Also important was all the support from Saskia, Robbert, Bart, and Ruud, who ensured Amogh was never stuck or lost on his journey. They get the ***Kindest Supporters Award***. Lastly, although the Awards Committee is not biased, it would like to mention that the computer vision lab is the best lab group ever and it's members are absolutely awesome - they get the prestigious ***Best Research Lab Award***. "Thank you guys for letting me pick your brains", remarks Amogh.

Although the PRB vision lab in Delft was where Amogh did his PhD in spirit, the VicarVision office at the SMR HQ in Amsterdam was where he was in person. This is where most of Amogh's PhD research projects were born, and these projects wouldn't have been so much fun without the people of VicarVision and SMR: Nicolai, Kasper, Tess, Paul, Peter, Marian, Tim, Lysbeth, George, Pieter, Hans, Franjo, Hans, Vincent, Roos and Ineke (in order of their desks' proximity to Amogh's). Each one of them gets the ***Incredible Colleague Award***. Previous PhD'ers deserve a special mention for lighting the spark of PhD in Amogh — Hans, Peter, Tess, Emrah and Andreas — they get the ***PhD Inspirer Award***. Finally, another special award goes to the CEO Tim den Uyl for his awesome leadership, and for continuing to support and seeing value in Amogh's research. He gets the coveted ***Best CEO Award***!

As we approach the end of this awards session, it is time to thank the amazing people around Amogh whom he had a chance to spend good times with, be it in Amsterdam, Delft, Bangalore, on Greek islands, on Austrian slopes, on road trips, in pubs, on boulder walls, or on App Groups (*$ANSSCAR*). All of these awesome people get the ***Super Friends Award***. Recognition also goes out to Prakash, who is awarded the ***OK Housemate Award***. Neha, of course, gets the ***Best Girlfriend Award*** for (i) agreeing to be the girlfriend of Amogh, and for (ii) her constant encouragement and fair criticism (Amogh would like to reassure Neha that he is *not* building *Skynet*).

Finally, Amogh's PhD wouldn't have been possible without the support of his family (including the whole *Gudi++* clan). Amogh's parents — Vinodini and Anirudh — have always encouraged him to go forward on this academic path, and their support and patience has been very important; they obviously get the ***Best Parents Award***. Apoorva gets the ***Annoying-but-Good Sister Award***, and Ishaan gets a high-five (also an award).

The Awards Committee hopes all the winners[1] have had a great time themselves during this PhD, and informs them that Amogh hopes to see them around in the years to come.

Amogh Gudi PhD Awards Committee
Amsterdam-Noord, October 2022

[1]Nominees who's mentions may have been missed in these prestigious awards may appeal to the Awards Committee. They shall be compensated with their favourite beverages.

1

Introduction

If we wait for the moment when absolutely everything is ready, we shall never begin.

Ivan Turgenev

1

Faces and facial expressions are prominent ways of human social communication [1]. Faces visually communicate not only the dynamic emotional and cognitive states of a person, but several other static aspects such as age, gender and ethnicity [2]. Several internal physiological states can also be measured from the face, such as heart rate [3] and blood pressure [4]. Body pose [5] and gesture [6] analysis constitute the other important components of human observation, since humans also give out social signals through these mediums [7]. With the advent of technology, interactions between humans and machines become more prevalent. These interactions can be more effective if machines are also able to interpret information from faces to understand humans. This thesis focuses on the application area of human observation and facial analysis using machine vision.

As a sub-field of Artificial Intelligence (AI), Machine/Computer Vision aims to let machines 'see': using computations on the input of visual sensors to detect humanly meaningful concepts. Machine vision system are able to automate several visual tasks such as: *detection/localization* — detecting presence of certain objects and locating them, e.g., face detection [8]; *recognition* — identifying the type of the presented object/scene, e.g., anomaly detection [9]; *measurement* — estimating the intensity of an underlying phenomenon, e.g., visual heart-rate estimation [10]; *modelling* — building a geometric model of the object/scene, e.g., face modelling [11]; *filtering/transformation* — modify the given visual data into another form, e.g., image restoration [12]; etc. Several of these topics are covered in the chapters of this thesis.

Machine vision system achieve automation of visual tasks either by relying on manual programming of the algorithm based on the designer's knowledge, or by learning the algorithm directly from available data. For machine vision systems to have any societal impact, they should work in the real world where the available resources are limited. These are resources used for computational execution/inference, but also information/data is a resource which is used for the design and training of the system. Figure 1.1 showcases the typical resources used by such a system. Machine vision systems need to be efficient in their use of these resources during their design as well as during their deployment. In this thesis, we explore machine vision techniques that maximize the utility gained by efficient use of these resources.

1.1 Resources in Machine Vision

1.1.1 Computational Resources

The number of computational operations available per second on the computing device is a resource for machine vision systems. Because only a limited number of operations can be performed in a given time: if fewer operations are required, the time spent by the system on computation reduces. Computation time is important for practical use of vision systems. For example, facial expression recognition by a robot should work in real-time, otherwise interaction with a human is impossible. Also for developing the methods themselves, it is important to have fast execution because it allows for quicker prototyping. Thus, reducing the number of computational operations is essential for real-world applications, as we explore in this thesis.

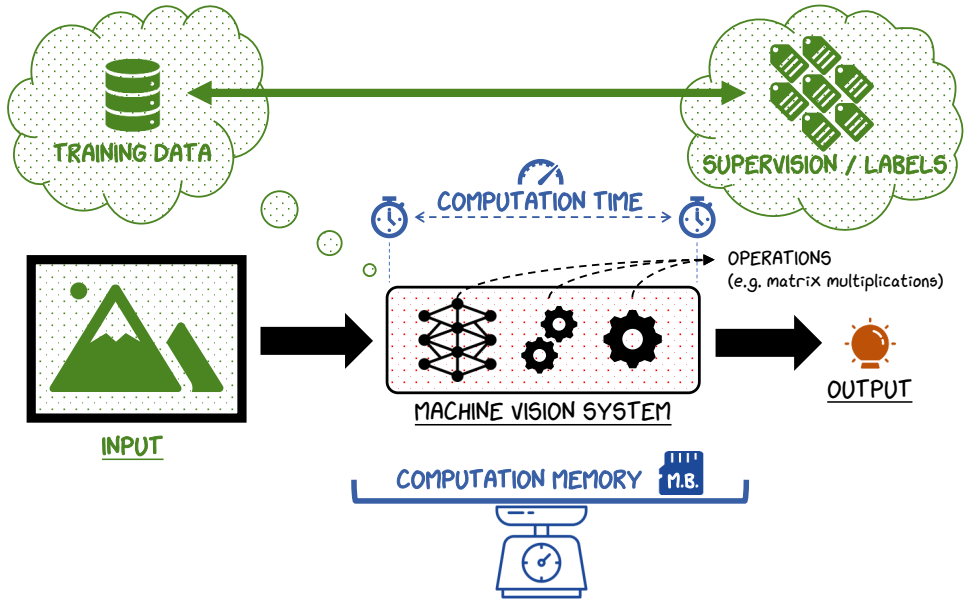


Figure 1.1: A diagram showing the various resources used by a machine vision systems. The resources shown in green represent informational resources, while blue represents computational resources. The machine vision system is designed/trained using collected training data, potentially along with their manual labels (supervision). This system then process the information in the provided input (also considered a resource) while expending computational resources (time and memory) to generate the required output.

Another computational resource is memory, which determines how many bytes can be used during execution. If fewer bytes are needed, then the machine vision system can be deployed on a wider range of devices because several devices have limited memory. For example, edge devices like smartphones may not be able to load recent face detectors on their limited memory. Also, high memory requirements limit what hardware can be used for design and development. For example, training deep neural networks requires high-end GPUs. Therefore, reducing the computational memory requirements has significant practical value.

Time and memory limitations can typically be solved by using more expensive high-end hardware. Yet, even for the exceptionally deep pockets of tech companies, this is not always feasible because of inherent restrictions in the use cases. For example, portable electronics need to run on low power from small batteries, and therefore their processing capabilities are low; mass-market consumer devices need to be low-cost for commercial viability, and this makes high-end processors unaffordable. For such real-world problems, developing computationally efficient systems is often the only viable solution.

1

1.1.2 Informational Resources

Machine learning has recently shown excellent results for vision tasks (e.g., AlexNet image classifier [13], Dall·E-2 image generator [14]). The parameters of machine learning based methods are determined by optimizing for a training dataset of input examples and their expected outputs. Typically, increasingly better parameters can be learned from increasingly larger training datasets. However, creating such datasets require significant manual effort: samples in the dataset must be collected, filtered and annotated with their expected outputs. Due to this manual effort in combination with increased storage needs, amassing large datasets is expensive. Therefore, it is often desirable to design systems that provide a high accuracy while requiring lower quantity of training data.

Concurrent to the quantity of training data, the quality/size of the data samples is also an informational resource. The quality and richness of the input data, i.e., the amount of information present in it, can make the task easier for the system leading to improved accuracy. For example, RGB color images are more information rich than grayscale images and this can be useful for classifiers. However, in several situations, access to such high quality input data may not be available. In other situations, the information held in the input could be redundant and distracting for the system to perform a given task. Designing methods to use pruned input to reduce the amount of non-vital information can improve the input-data efficiency of the system. For example: roughly the same facial expression recognition accuracies can be obtained from smaller 48×48 sized input images as larger 72×72 images [2]; downsizing a noisy image (from a cheap camera) can help reduce noise levels [15] which could have be distracting; cropping images to focus on known key regions can avoid processing of non-vital information [16]. As an additional consequence of reducing the input size, the system is required to do less computation, thereby lowering its computational resource needs as well.

Just as input data is a resource, the annotation labels associated with them are also a resource. These annotation labels are the source of supervision during the training phase of machine learning based systems, and supervision is typically vital for good performance. The quantity and quality of labels directly affect learning: the more abundant, cleaner and informative the labels are, the better the model can learn and perform. While the collection of training input samples itself can be an effort, manual labelling of these samples is typically far more laborious (see Figure 1.2 for an illustration). This becomes especially true when the labels are required to be fine-grained and noise-free (e.g., pixel-level semantic segmentation maps [17], facial action unit annotations [18]). Therefore, reducing the dependence on the quantity and quality of labels can greatly help bring down manual developmental effort.

1.2 Efficiency Methods for Machine Vision

1.2.1 Informationally Efficient Techniques

The computer vision systems that are crafted by hand are based on the knowledge held by the expert designer. Therefore, development of hand-crafted methods involving no machine learning do not require access to substantial training datasets. Due to this lack

of training data requirement, such hand-crafted prior-knowledge based systems can be considered very information efficient, specifically training-data quantity efficient. As a caveat, such methods are more dependant on the assumptions made in this prior knowledge. Incorrectness of such assumptions can lead to wrong design choices, which may not be easily undone.

At the same time, the prior domain knowledge that the system design depends upon is created based on the experts'/designers' experiences with/understanding of the task, and this *can* also be considered as a sort of information gathering. In addition, the designer may still need access to a relatively small dataset for iteratively testing the system through its design process. For example, in the design of signal processing pipelines for estimating heart rate from face videos, a data collection effort was still necessary to iteratively validate the method [3, 19, 20]. Thus, hand-crafted systems still have some amount of data requirements, albeit small, even in the absence of machine learning.

Hand-crafted techniques have dominated low and mid-level task in computer vision until very recently [21–23]. These include detectors like Canny edge detector [24], Harris corner detector [25] at the low-level, and feature descriptors like SIFT [26], SURF [27] at the mid-level. Techniques utilizing Gaussian filters and derivatives for scale-invariance are examples of hand-crafted methods based on scale-space theory [28] as their prior knowledge. These priors have also been incorporated into machine learning models such as convolutional neural networks by replacing its kernels with Gaussians [29, 30]. However, with the increase in the availability of training data and powerful computing hardware, learning based methods like deep learning have taken over as the state-of-the-art in several tasks. This thesis explores the incorporation of prior knowledge into machine vision systems to make them more efficient.

Fully supervised learning is the standard approach for training a learning-based model for a given task, which requires access to a labelled dataset. Typically for a given model, the larger the amount of data available for training, the better the performance. However, after a certain amount, increasing the size of the training set yields diminishing results. Therefore, a data-efficient way of performing supervised training would be to limit collection of training data to this point (e.g., using learning curves [31, 32]). Additional 'tricks' like data augmentation [33] can further maximise the utility of existing data. A popular approach is to generate synthetic data for training based on prior-knowledge driven data models (e.g., a graphics engine) [34]. These techniques can be used to make supervised learning function with a reduced quantity of training data.

For several vision tasks, a full resolution image of the whole scene as input may not be necessary/useful for the model. Pre-processing techniques such as image cropping can be performed to only select the relevant parts of the input. Additionally, the input image may be downscaled to make the model compatible with lower resolution images. While a majority of informational resources relate only to the training phase of learning-based systems, input-data concerns both: training as well as inference; since input is processed the same way in both phases.

Semi-supervised learning aims to reduce the labelling/annotation requirement of the training dataset. Predominantly during data collection in computer vision, input samples are far more easily available than manual labels for them. Semi-supervision takes advantage of the relatively high abundance of unlabelled input samples by also learning from them. This is achieved in practice through several methods such as un/self-supervised pre-training followed by supervised fine-tuning [35] and pseudo-labelling [36].

Un/Self-supervised pre-training works by first extracting knowledge by training the model to recreate (parts of) the unlabelled input itself [37], and later fine-tuning the pre-trained model to fit the available labelled data [38]. In contrast, pseudo-labelling works by first training on the labelled data, and using the trained model to assign pseudo-labels for the unlabelled data [39]. A more extreme form of semi-supervised learning is few/zero-shot learning, where the objective is to learn from only a handful (or none) training examples of a target domain/class, labelled or otherwise [40]. Generally for a given task and model, semi-supervised learning can obtain better results than fully supervised learning when the labelled training dataset size is limited. These techniques can therefore improve the label-efficiency of the system, and are studied in this thesis.

Apart from reducing the required quantity of the labelled data, another way to ease labelling effort is to lower the requirement on the quality/richness of the labels. Learning with such a weakened form of supervision is known as weakly-supervised learning. These weak labels are usually in the form of imprecise annotations (e.g., bounding boxes instead of segmentation maps) or inaccurate labels (e.g., crowdsourced labels) [41]. Figure 1.2 shows examples of strong and weak labels with varying degrees of supervision.

Weakly-supervised learning is implemented in several forms and several tasks such as object localization and segmentation. The strategies range from obtaining and combining multiple candidate outputs from the weakly-trained model to produce a more precise output [42, 43], to extracting fine-grained information from the internal activations of the model [44, 45]. We study ways to improve the efficieny of weakly-supervised methods in this thesis.

1.2.2 Computationally Efficient Techniques

The size and complexity of the models used in computer vision systems are the primary factors affecting the use of computational resources. Larger models perform a higher number of computational operations (e.g., matrix multiplications), thereby resulting in longer computation times. Larger models also contain a greater number of parameters, storing which consumes more memory. Therefore, utilizing smaller models with fewer parameters and operations in computer vision systems directly improves the overall computational efficiency.

However, it is not straightforward or trivial as to how model sizes can be kept small for a given task. Understanding and analysis of the underlying operations (such as convolutions) can provide hints as to how the model architecture can be optimized [46]. Regardless, discovery of such 'tricks' usually involves relatively large scale iterative hyper-parameter search experimentation. A significant body of research has contributed several design principles geared towards improving the computational efficiency of deep neural network

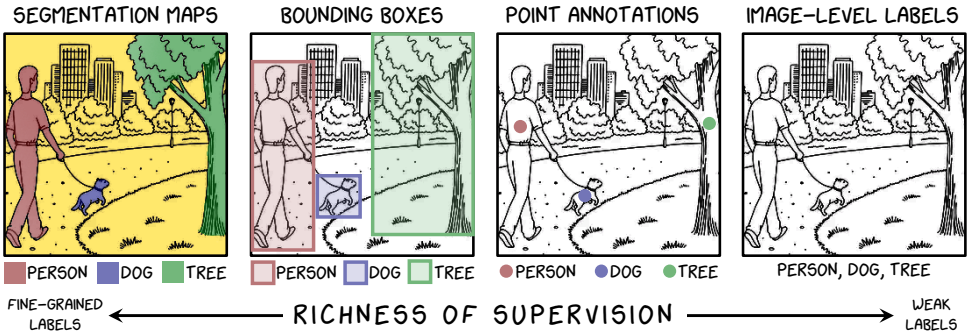


Figure 1.2: An illustration showcasing different levels of supervision used in the training of learning-based computer vision models, in this case for object localization. Rich/fine-grained labels such as pixel-wise segmentation maps are high in informational content, but require maximum labelling effort. Bounding boxes and point supervision require progressively less manual effort but also contain less information about the object's location. Image-level labels represent a weak form of supervision, containing minimal information but also requiring minimal effort. Label-efficient methods attempt to match richly supervised models using weaker labels.

models [47, 48]. These techniques essentially leverage upon known prior-knowledge about the task/mechanism to optimize the design of the model.

Alternatively, automated machine learning (AutoML) [49] techniques can automate the search for a computation-efficient architecture [50, 51] at the cost of additional design-time computation. Knowledge distillation [52] is another technique whereby the 'knowledge' learnt by a large 'teacher' model can be transferred into a smaller 'student' model [53]. Reducing the size of models can also be attempted with more direct approaches: model pruning [54, 55] and model quantization [56–58]; whereby redundant parameters are pruned, and their numeric precision is quantized down (e.g., 32-bit to 16-bit floats [56]). A caveat of these techniques is that they sometimes add to the computational expenses during training; even though they can eventually result in an efficient deployable model.

Computational-time efficiency of machine vision systems can also be improved by reducing the size of the input. This is because the smaller the input data, the fewer the number of computation operations needed to process it. For example, the image resolution in machine vision systems for real-time applications are often kept low to minimize computation-time. However, reducing image resolution excessively can also lead to loss of vital information leading to erroneous outputs. Nevertheless, input reduction is a simple solution for reducing computational needs, as studied in this dissertation.

1

1.3 Contributions

The research presented in this dissertation makes the following contributions towards improving the efficiency of deployable machine vision systems. A summary linking the presented work with the efficiency concepts discussed so far is illustrated in Figure 1.3. A majority of these contributions have been applied to facial analysis and human observation, and form parts of a commercially available software: FaceReader[1].

Chapter 2: Weakly Supervised Single-Shot Localization This chapter [45] presents a way of performing weakly supervised object detection in a computationally efficient manner. The proposed method includes an object extent pooling layer, that is able to capture the extent of the detected objects through weak image-level labels. Because of this, this method is the first of its kind to demonstrate weakly supervised localization in a single network pass, bringing the computational speeds on par with fully supervised single-shot methods. Thus, the presented work contributes towards improving the computational efficiency of an informationally efficient object detection technique.

Chapter 3: Input and Data Efficiency in Webcam Gaze Tracking
This chapter [16] investigates the role of contextual input information in predicting gaze direction from images; and calibration-data efficiency for projecting gaze onto a screen. The presented study weights up the computational cost of processing larger context-rich inputs against the accuracy gained from them. Additionally, this chapter analyzes the data efficiency of multiple screen calibration techniques in terms of the impact of prior knowledge. The results aid in improving input and computational efficiencies of webcam gaze tracking methods, as well as the data efficiency of gaze calibration techniques.

Chapter 4: Exploiting Spatial Context for Anomaly Detection & Feature Learning In contrast to the previous chapter, this work [59] attempts to maximize the gains from spatial context in image data. This chapter identifies the lack of locational sensitivity as a drawback of a commonly used distance function (mean squared error), and presents a proximally sensitive error function that takes the location of differences into consideration. Insights into the applicability of this error function is provided through informationally efficient implementations: few-shot learning for anomaly detection, and unsupervised pre-training for classification. The presented discussion suggests conditions under which the data, computation, and label requirements can be reduced using the proposed technique.

[1] `vicarvision.nl/products/facereader`; `noldus.com/facereader`

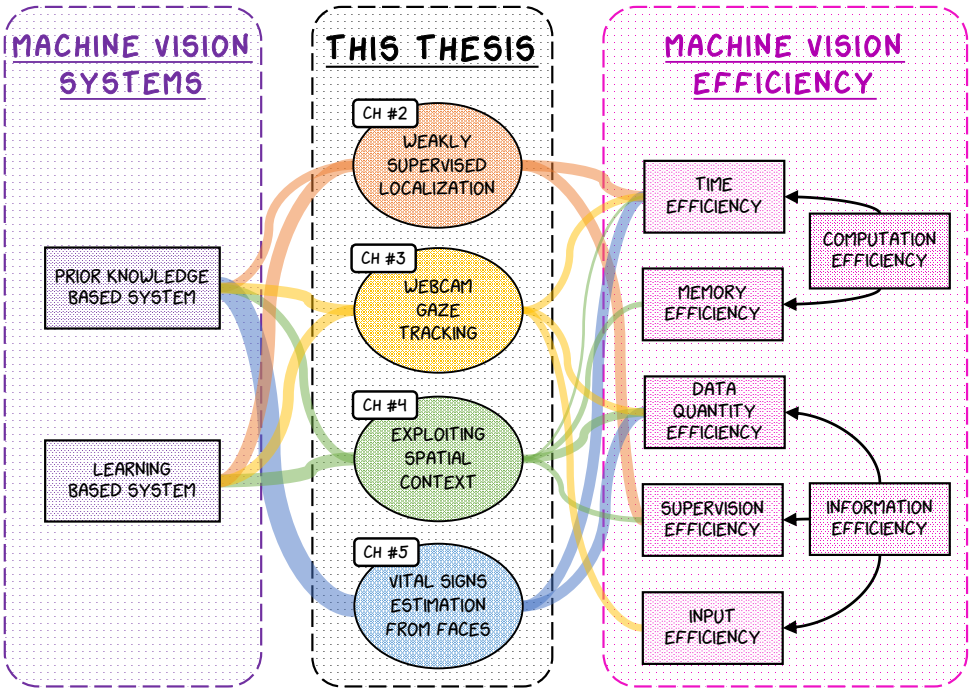


Figure 1.3: An illustration of the link between the presented work from this dissertation and the efficiency concepts in machine vision. The thickness of the coloured linking lines represents the strength of the connections. Chapters 3, 4, and 5 demonstrate applications in facial analysis. Incorporating prior knowledge is a common theme among all chapters, and this leads to improvement in computational and informational efficiency.

Chapter 5: Prior Knowledge driven Efficient Vital Signs Estimation from Faces
The last chapter [19, 20] presents an in-depth analysis of an unsupervised pipeline for measuring vital signs like heart rate and variability from faces. The presented work avoids learning from video data by relying upon prior knowledge, which greatly minimizes data collection effort. Also, the lack of computationally heavy components (like deep neural networks) makes this method computationally light, thereby enabling real-time use. Through an exhaustive study, this chapter shows how this unsupervised approach is able to surpass or match fully supervised methods. Low computational load and independence from training data makes this approach computationally and informationally efficient.

1

References

[1] R. E. Jack and P. G. Schyns, *The human face as a dynamic tool for social communication,* Current Biology **25**, R621 (2015).

[2] A. Gudi, *Recognizing semantic features in faces using deep learning*, Master's thesis, University of Amsterdam (2015), *arXiv preprint arXiv:1512.00743*.

[3] H. E. Tasli, A. Gudi, and M. den Uyl, *Remote ppg based vital sign measurement using adaptive facial regions,* in *2014 IEEE International Conference on Image Processing (ICIP)* (IEEE, 2014) pp. 1410–1414.

[4] I. C. Jeong and J. Finkelstein, *Introducing contactless blood pressure assessment using a high speed video camera,* Journal of medical systems **40**, 77 (2016).

[5] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, *Openpose: realtime multi-person 2d pose estimation using part affinity fields,* IEEE transactions on pattern analysis and machine intelligence **43**, 172 (2019).

[6] S. S. Rautaray and A. Agrawal, *Vision based hand gesture recognition for human computer interaction: a survey,* Artificial intelligence review **43**, 1 (2015).

[7] C. Müller, A. Cienki, E. Fricke, S. H. Ladewig, D. McNeill, and S. Tessendorf, *Body-language-communication,* An international handbook on multimodality in human interaction **1**, 131 (2013).

[8] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features,* in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, Vol. 1 (Ieee, 2001) pp. I–I.

[9] T. Ehret, A. Davy, J.-M. Morel, and M. Delbracio, *Image anomalies: A review and synthesis of detection methods,* Journal of Mathematical Imaging and Vision **61**, 710 (2019).

[10] W. Verkruysse, L. O. Svaasand, and J. S. Nelson, *Remote plethysmographic imaging using ambient light.* Optics express **16**, 21434 (2008).

[11] H. Van Kuilenburg, M. Wiering, and M. Den Uyl, *A model based method for automatic facial expression recognition,* in *European conference on machine learning* (Springer, 2005) pp. 194–205.

[12] D. Ulyanov, A. Vedaldi, and V. Lempitsky, *Deep image prior,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018) pp. 9446–9454.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks,* Advances in neural information processing systems **25**, 1097 (2012).

[14] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, *Hierarchical text-conditional image generation with clip latents,* arXiv preprint arXiv:2204.06125 (2022).

[15] H. Talebi and P. Milanfar, *Learning to resize images for computer vision tasks,* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021) pp. 497–506.

[16] A. Gudi, X. li, and J. van Gemert, *Efficiency in real-time webcam gaze tracking,* in *Computer Vision – ECCV 2020 Workshops*, Part of the Lecture Notes in Computer Science book series (LNCS, volume 12535); also part of the Image Processing, Computer Vision, Pattern Recognition, and Graphics book sub series (LNIP, volume 12535), edited by A. Bartoli and A. Fusiello (Springer, 2020) pp. 529 – 543, European Conference on Computer Vision (ECCV) 2020 Workshop on Eye Gaze in AR, VR, and in the Wild (OpenEyes), ECCVW 2020; OpenEyes 2020; Conference date: 23-08-2020 Through 28-08-2020.

[17] J. Long, E. Shelhamer, and T. Darrell, *Fully convolutional networks for semantic segmentation,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015) pp. 3431–3440.

[18] A. Gudi, H. E. Tasli, T. M. Den Uyl, and A. Maroulis, *Deep learning based facs action unit occurrence and intensity estimation,* in *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, Vol. 6 (IEEE, 2015) pp. 1–5.

[19] A. Gudi, M. Bittner, R. Lochmans, and J. van Gemert, *Efficient real-time camera based estimation of heart rate and its variability,* in *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019 (2019) pp. 1570–1579.

[20] A. Gudi, M. Bittner, and J. van Gemert, *Real-time webcam heart-rate and variability estimation with clean ground truth for evaluation,* Applied Sciences **10**, 1 (2020).

[21] S. Agaian, A. Almuntashri, and A. Papagiannakis, *An improved canny edge detection application for asphalt concrete,* in *2009 IEEE International Conference on Systems, Man and Cybernetics* (IEEE, 2009) pp. 3683–3687.

[22] J. J. Anitha and S. Deepa, *Tracking and recognition of objects using surf descriptor and harris corner detection,* International Journal of Current Engineering and Technology **4**, 775 (2014).

[23] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, *Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding,* in *2010 IEEE International Conference on Robotics and Automation* (IEEE, 2010) pp. 2308–2315.

[24] J. Canny, *A computational approach to edge detection,* IEEE Transactions on pattern analysis and machine intelligence , 679 (1986).

[25] C. Harris, M. Stephens, *et al.*, *A combined corner and edge detector,* in *Alvey vision conference*, Vol. 15 (Citeseer, 1988) pp. 10–5244.

[26] D. G. Lowe, *Object recognition from local scale-invariant features,* in *Proceedings of the seventh IEEE international conference on computer vision*, Vol. 2 (IEEE, 1999) pp. 1150–1157.

[27] H. Bay, T. Tuytelaars, and L. Van Gool, *Surf: Speeded up robust features,* in *European conference on computer vision* (Springer, 2006) pp. 404–417.

[28] T. Lindeberg, *Scale-space theory: A basic tool for analyzing structures at different scales,* Journal of applied statistics **21**, 225 (1994).

[29] J.-H. Jacobsen, J. Van Gemert, Z. Lou, and A. W. Smeulders, *Structured receptive fields in cnns,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 2610–2619.

[30] N. Saldanha, S. L. Pintea, J. C. van Gemert, and N. Tomen, *Frequency learning for structured cnn filters with gaussian fractional derivatives,* in *British Machine Vision Conference (BMVC)* (2021).

[31] F. Mohr and J. N. van Rijn, *Learning curves for decision making in supervised machine learning–a survey,* arXiv preprint arXiv:2201.12150 (2022).

[32] T. Viering and M. Loog, *The shape of learning curves: a review,* arXiv preprint arXiv:2103.10948 (2021).

[33] C. Shorten and T. M. Khoshgoftaar, *A survey on image data augmentation for deep learning,* Journal of Big Data **6**, 1 (2019).

[34] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, *Playing for data: Ground truth from computer games,* in *European conference on computer vision* (Springer, 2016) pp. 102–118.

[35] A. Coates, A. Ng, and H. Lee, *An analysis of single-layer networks in unsupervised feature learning,* in *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (JMLR Workshop and Conference Proceedings, 2011) pp. 215–223.

[36] D.-H. Lee *et al.*, *Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,* in *Workshop on challenges in representation learning, ICML*, Vol. 3 (2013) p. 896.

[37] L. Jing and Y. Tian, *Self-supervised visual feature learning with deep neural networks: A survey,* IEEE transactions on pattern analysis and machine intelligence (2020).

[38] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin, *Unsupervised pre-training of image features on non-curated data,* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019) pp. 2959–2968.

[39] N. F. Chen, *Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion,* in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2018) pp. 644–653.

[40] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, *Generalizing from a few examples: A survey on few-shot learning,* ACM Computing Surveys (CSUR) **53**, 1 (2020).

[41] Z.-H. Zhou, *A brief introduction to weakly supervised learning,* National science review **5**, 44 (2018).

[42] H. Bilen and A. Vedaldi, *Weakly supervised deep detection networks,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 2846–2854.

[43] C. Wang, W. Ren, K. Huang, and T. Tan, *Weakly supervised object localization with latent category learning,* in *European Conference on Computer Vision* (Springer, 2014) pp. 431–445.

[44] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, *Learning deep features for discriminative localization,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) pp. 2921–2929.

[45] A. Gudi, N. Van Rosmalen, M. Loog, and J. Van Gemert, *Object extent pooling for weakly supervised single-shot localization,* in *British Machine Vision Conference 2017, BMVC 2017*, British Machine Vision Conference 2017, BMVC 2017 (BMVA Press, 2017) 28th British Machine Vision Conference, BMVC 2017 ; Conference date: 04-09-2017 through 07-09-2017.

[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015) pp. 1–9.

[47] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, *Mobilenets: Efficient convolutional neural networks for mobile vision applications,* arXiv preprint arXiv:1704.04861 (2017).

[48] J. Lin, C. Gan, and S. Han, *Tsm: Temporal shift module for efficient video understanding,* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019) pp. 7083–7093.

[49] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, *Auto-weka: Combined selection and hyperparameter optimization of classification algorithms,* in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013) pp. 847–855.

[50] M. Tan and Q. Le, *Efficientnet: Rethinking model scaling for convolutional neural networks,* in *International Conference on Machine Learning* (PMLR, 2019) pp. 6105–6114.

[51] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, *Amc: Automl for model compression and acceleration on mobile devices,* in *Proceedings of the European conference on computer vision (ECCV)* (2018) pp. 784–800.

[52] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network,* in *NIPS Deep Learning and Representation Learning Workshop* (2015).

[53] J. Gou, B. Yu, S. J. Maybank, and D. Tao, *Knowledge distillation: A survey,* International Journal of Computer Vision **129**, 1789 (2021).

[54] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, *What is the state of neural network pruning?* arXiv preprint arXiv:2003.03033 (2020).

[55] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, *Pruning convolutional neural networks for resource efficient inference,* in *5th International Conference on Learning Representations, ICLR - Conference Track Proceedings* (2017).

[56] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, *Bag of tricks for image classification with convolutional neural networks,* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019) pp. 558–567.

[57] M. Nagel, M. v. Baalen, T. Blankevoort, and M. Welling, *Data-free quantization through weight equalization and bias correction,* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019) pp. 1325–1334.

[58] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, *Binary neural networks: A survey,* Pattern Recognition **105**, 107281 (2020).

[59] A. Gudi, F. Büttner, and J. van Gemert, *Proximally sensitive error for anomaly detection and feature learning,* in *ArXiv:2206.00506; Extended Abstract, ICT.OPEN, Hilversum* (2019).

2

Weakly Supervised Single-Shot Localization

In the face of scarcity in detailed training annotations, the ability to perform object localization tasks in real-time with weak-supervision is very valuable. However, the computational cost of generating and evaluating region proposals is heavy. We adapt the concept of Class Activation Maps (CAM) into the very first weakly-supervised 'single-shot' detector that does not require the use of region proposals. To facilitate this, we propose a novel global pooling technique called Spatial Pyramid Averaged Max (SPAM) pooling for training this CAM-based network for object extent localisation with only weak image-level supervision. We show this global pooling layer possesses a near ideal flow of gradients for extent localization, that offers a good trade-off between the extremes of max and average pooling. Our approach only requires a single network pass and uses a fast-backprojection technique, completely omitting any region proposal steps. To the best of our knowledge, this is the first approach to do so. Due to this, we are able to perform inference in real-time at 35fps, which is an order of magnitude faster than all previous weakly supervised object localization frameworks.

This chapter is based on *Gudi, A., Van Rosmalen, N., Loog, M., & Van Gemert, J. (2017). Object extent pooling for weakly supervised single-shot localization. In British Machine Vision Conference 2017, BMVC 2017 (British Machine Vision Conference 2017, BMVC 2017). BMVA Press.* https://doi.org/10.5244/c.31.36 [1].

2.1 Introduction

Weakly supervised object localization methods [2, 3] can predict a bounding box without requiring bounding boxes at train time. Consequently, such methods are less accurate than fully-supervised methods [4–7]: it is acceptable to sacrifice accuracy to reduce expensive human annotation effort at *train time*. Similarly, blazing fast fully supervised single-shot object localization methods such as YOLO [7] and SSD [6] make a similar trade-off of running speed versus accuracy at *test time*. More accurate methods [4, 5] are slower and thus exclude real-time embedded applications on a camera, drone or car. In this chapter, we optimize for speed at train time and at test time: We propose the first weakly supervised single-shot object detector that does not need expensive bounding box annotations during train time and also achieves real-time speed at test time.

Exciting recent work has shown that object detectors emerge automatically in a CNN trained only on global image labels [8–10]. Such methods convincingly show that a standard global max/average-pooling of convolutional layers retain spatial information that can be exploited to locate discriminative object parts. Consequently, they can predict a point inside the ground truth bounding box with high accuracy. We take inspiration from these works and train only for image classification while exploiting the spatial structure of the convolutional layers. Our work differs in that we do not aim for predicting a single point inside the bounding box, we aim to predict full extent of the object: the bounding box itself.

For predicting the object's extent, we have to decide how object parts are grouped together. Different object instances should be separated while different parts of the same object should be grouped together. Successful state-of-the-art methods on object localization have therefore incorporated a local grouping step in the form of bounding box proposals [4, 5]. After grouping, it is enough to indicate object presence and the object localization task is simplified to a bounding box classification task. In our work, we use no bounding boxes during training nor box proposals during testing. Instead, we let the CNN do the grouping directly by exploiting the pooling layer.

The pooling in a CNN groups pixels in a high-resolution image to a lower resolution one. Choices in pooling determine how the gradient is propagated back through the network. In average-pooling, the gradient is shared over all underlying pixels. In the case of a global

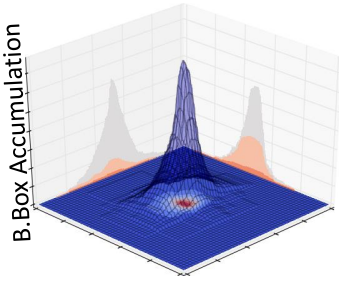


Figure 2.1: Accumulation of ground truth bounding boxes of Pascal VOC 2007 centered at the object's maximum activation. Note that the average extent follows a long-tailed distribution.

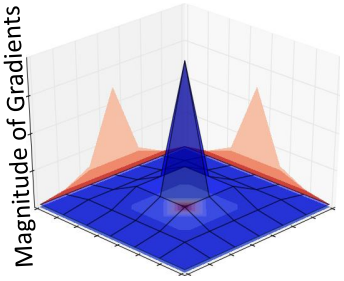


Figure 2.2: Gradient flow from our region pooling layer centered around the max activation. Note that our pooling follows the average extent illustrated in Figure 2.1.

image label, average-pooling will propagate loss gradients to all pixels in the image equally, which will cover the object but will also cover the background. In contrast, max-pooling only promotes the best point and will thus enforce only a single discriminative object part and not the object extent. Average-pooling is too wide, and max-pooling is too narrow; a regional pooling is needed for retaining the extent. Consider Fig 2.1, where we center the ground truth bounding boxes around its most discriminative part, given by the maximum filter response [9]. The average object extent is peaked, but has heavy tails. This motivates the need for regional pooling. In Fig 2.2, we show the gradient flow of our proposed pooling method centered around the maximum response. Our pooling method not only assigns gradients to the maximum or to the full image: it pools regionally.

We present the very first weakly-supervised single-shot detector. It has the following novelties. (i) **Speed**: we extend the idea of class activation maps (CAM) [10] onto a single stage CNN-only architecture for weakly supervised object localization, that achieves good accuracy while being 10-15 times faster than other related methods. (ii) **Extent pooling**: a 'regional' global pooling technique called the Spatial Pyramid Averaged Max (SPAM) pooling for capturing the object extent from weak image-level labels during training. (iii) **No region proposals**: We demonstrate a simple and fast back-projection pipeline that avoids the need for costly region proposal algorithms [11]. This allows our framework to perform real-time inference at 35fps on a GPU.

2.2 Related Work

Fully Supervised Object Localization. The state of the art is based on the R-CNN [12] pipeline which CNN combines the power of a classification network (e.g. ResNet [13]) with an SVM classifier and unsupervised region proposals [11]. This idea was sped up by [14] and [15] and many different algorithms emerged trying to propose the best regions [16–18], including a fully convolutional network [19] based version called R-FCN [4]. Recently published object detectors [6, 7] achieved orders of magnitude faster inference speeds with good accuracies by leaving region-proposals behind and predict bounding boxes in a single-shot. The high speed of our method is borrowed from the single-shot philosophy, albeit without requiring full supervision.

Weak Supervised Object Localization. Most methods [2, 3, 20, 21] follow a strategy where first, multiple candidate object windows are extracted using unsupervised region proposals [11], from each of which feature vector representations are calculated, based on which an image-label trained classifier selects the proper window. In contrast, our single-shot method does away with region proposals all together by directly learning the object's extent.

Li *et al.* [2] sets the state-of-the-art in this domain. They achieve this by filtering the proposed regions in a class specific way, and using MIL [22] to classify the filtered proposals. Bilen *et al.* [3] achieves similar performance by using an ensemble of two-streamed deep network setup: a region classification stream, and a detection steam that rank proposals. Wang *et al.* [20] starts with the selective search algorithm to generate region proposals, similar to R-CNN. They then use Probabilistic Latent Semantic Analysis (pLSA) [23] to cluster CNN-generated feature vectors into latent categories and create a Bag of Words

(BoW) representation to classify proposed regions. The work of Cinbis *et al.* [21] uses MIL with region proposals. In our work, we also are weakly-supervised, however, we perform localization in an end-to-end trainable single-pass without using region proposals.

A recent study by [9] follows an alternate approach [24] of using global (max) pooling over convolutional activation maps for weakly supervised object localization. This was one of the first works to use this approach. Their method gives excellent result for predicting a single point that lies inside an object, while predicting its bounding boxes, via selective search region proposals, yields limited success. In our work, we focus on ascertaining the bounding box extent of the object directly. Further efforts by [8] improve upon [9] in bounding box extent localization by using a tree search algorithm over bounding boxes derived from all final layer CNN feature maps. In our work, we perform extent localization of an object by filtering CNN activations into a single feature map instead of using a search algorithm, which makes our approach faster and computationally light, achieving high-speed inference.

Finally, the concept of class activation mappings in [10] serves as a precursor to our architecture. Like us, they make the observation that different global pooling operations influence the activation maps differently. We build upon their work and introduce object extent pooling.

2.3 Method

To allow weak supervision training for localization for a convolutional-only neural network, we use a training framework ending in a convolutional layer with a single feature map (per object class). This is followed by a global pooling layer, which pools the activation map of the previous layer into a single scalar value, which depends on the pooling method. This output is finally connected to a two-class softmax cross-entropy loss layer (per class). This network setup is then trained to perform image classification by predicting the presence/absence of objects of the target class in the image using standard back-propagation using image-level labels. A visualization of this setup is shown in Figure 2.3.

During inference, the global pooling and the softmax loss layers are removed, thereby the single activation map of the added final convolutional layer becomes the output of the network, in the form of an $N \times N$ grid. Due to the flow of backpropagated gradients through the global pooling layer during training, the weights of this convolutional layer get updated such that the location and shape of the strongly activated areas in its activation map essentially have a one-to-one relation with the location and shape of the pixels occupied by positive class objects in the image. At the same time, the intensity of the activation values in this activation map essentially represent the confidence of the network about the presence of the objects at the specific location. Borrowing notation from [10], we call this single feature-map output activation a Class Activation Map (CAM).

Consequently, to extract the location of the object in the image, the CAM activations are thresholded and backprojected onto the input image to localize the positive class objects.

2.3.1 The Class Activation Map (CAM) Layer

The class activation map layer is essentially a simple convolutional layer, albeit with a single feature map/channel (per object class) and a kernel size of 1×1. When connected to the

final convolutional layer of a CNN, the CAM layer has one separate convolutional weight for each activation map of the previous layer (see Figure 2.3). Training the network under weak-supervision through global pooling and softmax loss updates these kernel weight of the CAM layer through the gradients backpropagated from the global pooling layer. Eventually, the feature maps (of the previous conv layer) that produce useful activations for the training task of presence/absence classification are weighted higher, while the feature maps whose outputs are uncorrelated with the presence/absence of the positive class objects are weighted lower. Hence, the CAM output can be seen as the weighted sum combination of the activations of all the feature maps of the previous convolutional layer. Finally after training, the CAM activation essentially forms a heatmap of location likelihood of positive class objects in the input image.

The CAM layer used here is based on the concept of class activation mapping introduced in [10]. While being algorithmically similar, it should be noted that our CAM layer setup is different from the one in [10] in the following way: we perform the global pooling operation *after* the weight multiplication step (via a 1×1 conv.), while [10] does this *before* the weight multiplication step (via a FC layer). The reason for this difference is to allow greater ease of implementation and lower computational redundancy (requiring pooling on just one feature map).

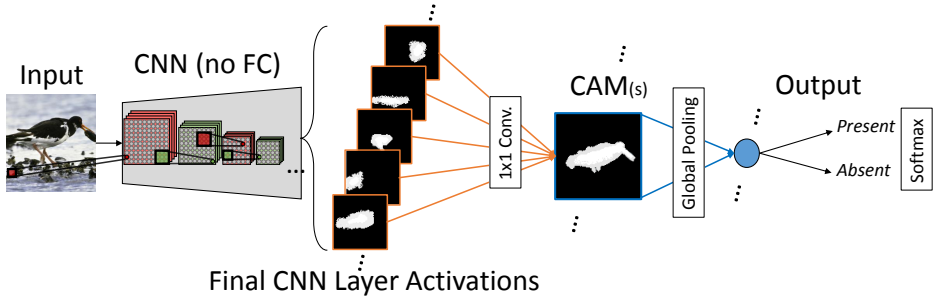


Figure 2.3: Visualization of the training setup for a CAM-augmented CNN. An extra conv. layer with a single feature map, the CAM, extracts the relevant feature information from the CNN's last conv layer. For weakly supervised training with present/absent annotation, the CAM is followed by a global pooling layer and connected to a softmax output/loss layer.

Inference

The complete pipeline is illustrated in Figure 2.4. A peak of CAM's activations would occur at the location corresponding to the most discriminative part of the object. The height of the peak is related to network confidence, whereas the extent of the object is captured by the width. To get a localization proposal, we can investigate which pixels in the original image where responsible for the activations that form a peak in the CAM. First, only the CAM peaks above the CAM threshold (computed based on the ratio of biases/weights of the output layer, learnt during training) are considered. Next, using a floodfill algorithm, all activated pixels belonging to the 'mountain' of this peak (including those below the threshold) are selected, as illustrated on the central plot in Figure 2.4. These pixels are then backprojected onto the input image via a fast-backprojection technique explained in

Algorithm 1: Fast-backprojection

Input: [X], [Y], $\text{layer}_{\text{CAM}}$, r // activation pixels in CAM layer, the CAM layer, resize ratio
Output: bpImage // backprojection on input image
/* for each activation pixel in the CAM layer */
foreach $\{x, y\}$ *in* {[X],[Y]} **do**
  $x_0 = x_1 \leftarrow x$; $y_0 = y_1 \leftarrow y$; $l \leftarrow \text{layer}_{\text{CAM}}$ // init
  /* loop through all layers from CAM to input */
  while $l \neq layer_{input}$ **do**
    /* s, p, k = stride, padding, kernel size */
    $\{x, y\}_0 \leftarrow \{x, y\}_0 \times s - p$
    $\{x, y\}_1 \leftarrow \{x, y\}_1 \times s - p + k - 1$
    $l \leftarrow \text{layer}_{\text{CAM}-1}$ // Go to next layer
  /* If ratio is provided, correct locations */
  if $r \neq 0$ **then**
    $\{x, y\}_0 \leftarrow \{x, y\}_0 + (\{x, y\}_1 - \{x, y\}_0) \times r/2$
    $\{x, y\}_1 \leftarrow \{x, y\}_1 + (\{x, y\}_1 - \{x, y\}_0) \times r/2$
  $\text{bpImage}[y_0 : y_1, x_0 : x_1] = 1$ // fill bpImage

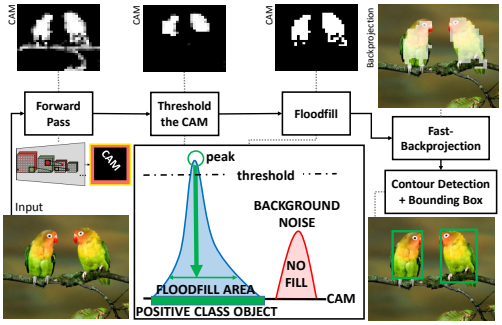


Figure 2.4: Visualization of the full inference pipeline. The central plot explains the thresholding and floodfilling steps. The outputs of the pipeline are positive class object bounding boxes.

Algorithm 1. We call it 'fast' because it computes the mapping between CAM pixels and the input pixels without actually performing a backward pass through the network. As can be inferred, this algorithm backprojects onto all pixels in the input image that could have contributed to the CAM activations (its receptive field). Therefore, we use a ratio parameter r to influence the size of the backprojected area. This parameter can be set by heuristics, or optimised over a separate validation set. Finally, by performing a contour detection on this backprojection, we can fit simple rectangular bounding boxes on the detected contours to localize the extent of the object.

2.3.2 Global Pooling

During training, the gradients computed from the loss layer reach the CAM layer through the global pooling layer. The connecting weights between the CAM and the previous conv layers are updated based on the distribution/flow of the gradients defined by the type of global pooling layer used. Hence, the choice of global pooling layer and its distribution of gradients to bottom layers is an important consideration for this framework for weak supervision.

Equation Legend In the equations hereafter, we consider a CAM activation map of $N \times N$, where x_n is an arbitrary pixel in it. The backpropagated gradients from the top loss layer is denoted by g.

Max and Average Pooling (GMP & GAP)

Global Max Pooling (GMP) layer is essentially a simple max pooling layer commonly used in CNNs, albeit whose kernel size is the same as the input image size. During the forward pass, this essentially means it always returns a single scalar pixel whose value is equal to the pixel with the highest value in the input image. During the backward pass, Equation 2.1 depicts how the gradients (∇_{GMP}) are computed for all pixel locations in the CAM layer.

It can be seen from the equation that the gradient is passed only to the location with the maximum activation in the CAM. During training with a positive object image, this implies that the detectors that additively contributed in making this pixel value high are encouraged via a positive weight update. Conversely, for a negative object image, the detectors that contributed in creating the highest value in the CAM are discouraged. Therefore, the network only learns from the image area that produces max activation in the CAM, i.e., the most discriminative object parts.

$$\nabla_{GMP} = g \cdot \begin{cases} 1, & \text{if } x_n = \max\limits_{0<=n<N}(x_n) \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

Global Average Pooling (GAP) layer performs a similar global pooling such that the single output pixel is the average of all input pixels during the forward pass. During the backward pass, the gradients are computed as denoted in Equation 2.2.

$$\nabla_{GAP} = g \cdot \frac{1}{N^2} \quad (2.2)$$

It can be seen that every location in the CAM gets the same gradient. Due to this, over multiple epochs of training, the detectors that fire for parts of the positive class object are strongly weighted, while detectors that fire for everything else are weighted very low. Thus, the network learns from all input image locations with an equal rate due to GAP's uniform backpropagated gradient.

The visualization of the gradient flow through these pooling layers is shown in Figure 2.5. Due to the single-location max-only gradient distribution of the global max pooling layer, it can be hypothesised that a GMP trained CAM can be quite ideal at pointing to the discriminative parts of an object. Conversely, due to the equally spread gradient distribution of the global average pooling layer, a CAM trained with GAP would activate for the full body of object plus parts of correlated or closely situated background.

Spatial Pyramid Averaged Max (SPAM) Pooling

Based on the properties of the global max and average pooling layers and from a study of pooling published in [25], we propose a pooling layer that is more tuned for training a CAM network for extent localization under weak supervision.

The approach consists of multiple local average pooling operations on the CAM activation map in parallel with varying kernel sizes. The kernel size of these average pooling operations is increased in steps (e.g., 1, 2, 4, ...), thus forming a spatial pyramid of local average pooling activation maps. Next, these activation maps are passed through global max pooling operations, which selects the maximum values among these average pooled activation maps. Finally, the output single pixel values of these combined pooling operation are averaged together to form the single scalar output of this layer. Due to the spatial pyramid structure and the use of average and max pooling operations, we call this layer global Spatial Pyramid Averaged Max Pooling, or simply SPAM pooling layer. A visualization of the architecture of SPAM layer is shown in Figure 2.6.

During the backward pass, the gradients are computed as depicted in Equation 2.3. Here, we consider a SPAM layer with P pyramid steps, each having a local average pooling

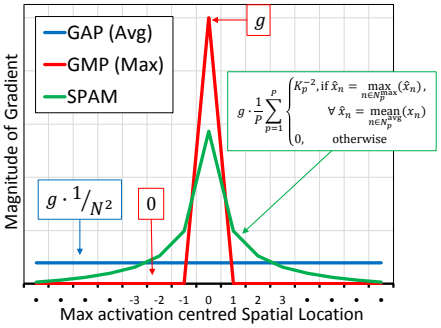


Figure 2.5: Visualization of gradient flow through global pooling layers. g is the backpropagated gradient from the upper later. The CAM size considered here is $N \times N$, and centered around its highest activation. SPAM pooling is considered to have P pyramid step, each with an average pooling kernel size of $K_p \times K_p$.

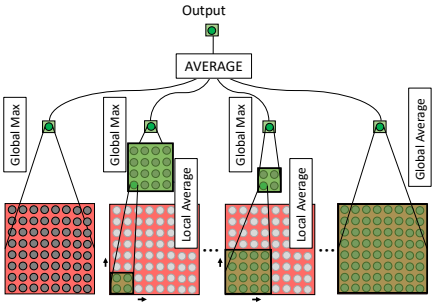


Figure 2.6: Architecture of the SPAM layer. First, local average pooling operations are applied in parallel with different kernel sizes, forming a pyramid of output activations. Next, global max pooling is applied and finally, its outputs are averaged. At the ends of the spatial pyramid, we directly show the equivalent GMP and GAP steps.

kernel size of $K_p \times K_p$; the backpropagated gradients from the top loss layer is represented g.

$$\nabla_{SPAM} = g \cdot \frac{1}{P} \sum_{p=1}^{P} \begin{cases} {K_p}^{-2}, & \text{if } \hat{x}_n = \max\limits_{n \in N_p^{\max}} (\hat{x}_n), \forall \hat{x}_n = \text{mean}\limits_{n \in N_p^{\text{avg}}} (x_n) \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

where the average/max pool kernel size at pyramid step p is $N_p^{\text{avg/max}} \times N_p^{\text{avg/max}}$.

The detectors responsible for creating maximal activation receives the strongest update, while the areas surrounding it receive an exponentially lower gradient that is inversely proportional to its distance from the maximal activation. As a result, while it strongly updates the weights of detectors of discriminative parts responsible for maximal activation, similar to GMP, it still ensures all locations receive a weak update, like in GAP. Due to this property, SPAM layer forms a good middle ground between the extremes of GMP and GAP. This can also be seen in Figure 2.5, which shows the gradients of SPAM layer, in comparison with that of global max and average pooling layers.

The gradient distribution of the SPAM layer is also shown in 3D in Figure 2.2, in comparison with the distribution of ground truth bounding boxes w.r.t the object's most discriminative part (given by CAM's maximal activation). As can be seen, SPAM's gradients are able to match the distribution of the objects' actual extent.

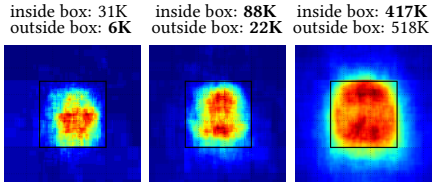
2.4 Experiments and Results

2.4.1 Evaluation of Global Pooling strategies on MNIST128

Setup As a proof of concept, we conduct experiments on a modified MNIST [26] dataset: MNIST128. this set consists of 28×28 MNIST digits placed randomly on a blank 128×128 image, thus creating a localization task. Further, we convert the 10-class MNIST classification problem to a two-class task where the digit 3 (chosen arbitrarily) is considered the positive

Method	mean Average Precision		
	Classification	Pin-pointing	Extent
GMP (Max)	**99.8**	**98.9**	69.5
GAP (Avg)	**99.4**	82.3	79.1
SPAM	**99.9**	**95.8**	**95.8**

Table 2.1: Results of the pooling experiments on MNIST128. Bold entries are the ones that perform 'well' on the two-class task (>95 mAP).



(a) GMP (b) SPAM (c) GAP

Figure 2.9: Visualization of the sum of normalized CAM activations, such that the object size present in the image is constant (denoted by the black box). The numbers denote the quantity of activated pixels (correctly) inside vs (wrongly) outside the objects' bounding box.

class, and rest are negative. We consider three types of tasks: classification, bounding box localization with at least 0.5 IoU (detection/extent localization), and localization by pin-pointing. Pin-pointing is identifying any single point that falls within the object bounding box [9]. We use a FC-less version of LeNet-5 [27] with our CAM extension, trained with softmax loss via various global pooling techniques. The SPAM pooling layer used here consists of a spatial pyramid of 4 steps, with local average pool kernel sizes 1×1, 2×2, 5×5, and $N \times N$, where N is the size of the CAM activation map. After training, the layers succeeding the CAM were removed, and inference was performed as explained in 10.

The results of this experiment are in Table 2.1. As hypothesised, GMP is good at locating the most discriminative part of the object, and thus succeeds at pin-pointing, but fails at extent. In comparison, GAP performs worse in pin-pointing, and better in extent. The global SPAM pooling is actually able to perform fairly better overall than both the other forms of pooling for object localisation.

2.4.2 Experiments on PASCAL VOC

Setup We adapted an ImageNet pre-trained version of VGG-16 [28]. We replaced the fully connected layers with our CAM layer, followed by our global SPAM pooling layer plus softmax output layer. Once again, the SPAM pooling used here consisted of 4 pyramid steps with kernel sizes of 1×1, 2×2, 5×5, and $N \times N$, where N is the size of the CAM activation map. To train our CAM layer weakly on the PASCAL VOC 2007 training set, we assigned a CAM-SPAM-softmax setup, see Fig 2.3, to each of the 20 VOC classes. After the training, we removed the layers succeeding the CAMs, as was done in the previous experiment. We also fine-tuned the ratio parameter in Algorithm 1 on a separate validation set.

Analysis of CAM behaviour trained via various Global Pooling techniques

To investigate our method further, we normalize and sum the CAM activations over the whole test set (only images contained one object), such that the size of the object in all the images is constant and centered. In Figure 2.9, we visualize the distribution of CAM's activated pixels w.r.t the object bounding box.

Figure 2.9 illustrate that the GMP trained CAM activations strongly lie within the

Method	mAP
PASCAL VOC 2007 test set	
SPAM-CAM[Ours]	**27.5**
GMP-CAM (Max Pool)[Ours]	25.9
GAP-CAM (Avg Pool)[Ours]	15.6
Li[RP+MIL] [2]	**39.5**
Bilen[RP+Ensemble] [3]	39.3
Wang[RP+pLSA] [20]	30.9
Cinbis[RP+MIL] [21]	30.2
Bency[RP+TreeSearch] [8]	25.7
PASCAL VOC 2012 validation set	
SPAM-CAM[Ours]	**25.4**
GMP-CAM (Max Pool)[Ours]	22.6
GAP-CAM (Avg Pool)[Ours]	19.3
Bency[RP+TreeSearch] [8]	**26.5**
Oquab[RP+GMP] [9]	11.7

Table 2.2: Detection results on PASCAL VOC 2007 & 2012. Entries marked with [RP] denote their use of region proposal sets.

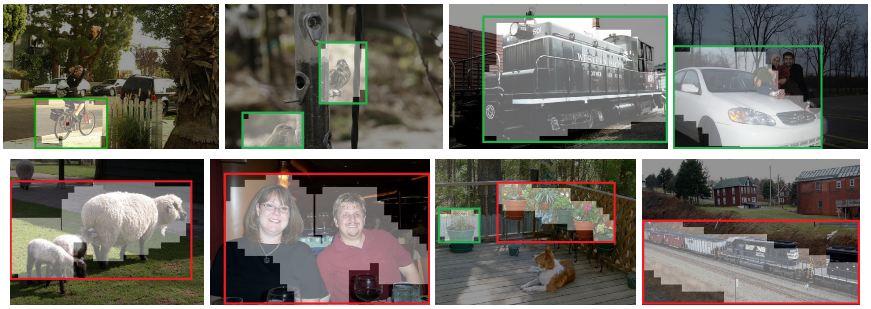


Figure 2.10: Localization examples: The highlighted areas in the images indicate the backprojection of CAM activations; green b.boxes match the ground truth, while red do not. Note how wrong b.box predictions are mostly either due to closely occurring objects, or closely correlated background.

bounding region of the object, but fail to activate for the full extent of the object. Conversely, GAP trained CAM activations spread well beyond the bounds of the object. In contrast, the activations of SPAM trained CAM do not spread much beyond the object's boundaries, while still activating for most of the extent of the object. This observations support our hypothesis that SPAM pooling offers a good trade-off between the adverse properties of GMP and GAP, and hence are better suited for training CAM for weakly supervised localization.

Comparison with the State of the Art

The results obtained with this network can be found in Table 2.2, in comparison with prior work. While evaluating these results, it should be noted that all the previous work in this field rely on region proposals, which is an extra computationally heavy step. [2] uses a combination of region proposals, multiple instance learning and fine-tuned deepnets, and [3] uses region proposals and an ensemble of three deep networks to achieve this

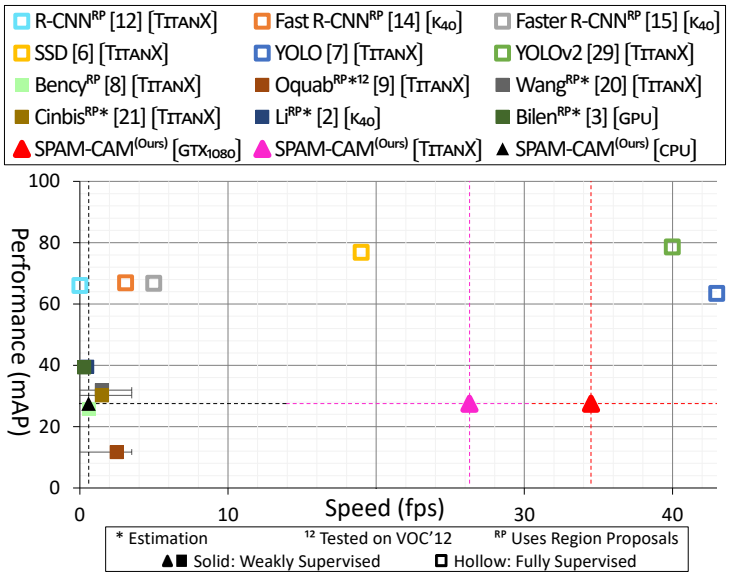


Figure 2.11: Speed and performance comparison between different localization methods on PASCAL VOC 2007 test set.

performance. In contrast, our method is purely single-shot, i.e., it requires a single forward pass of the whole image without the need of region proposals, which makes the method computationally very light. To the best of our knowledge, this is the first method to perform WSOL without region proposals.

Here, we see that the best methods [2, 3] using proposals perform significantly better. However, we are able to match the performance of other methods that also use region proposals [8, 9, 20, 21] and rely on similarly sized CNNs as ours. This observation suggests that region proposals themselves are not vital for the task of weakly supervised localization.

Speed Comparison In Figure 2.11, the performance of several methods is shown against the speed at which they can achieve this performance (on the PASCAL VOC 2007 test set). The test speeds for all methods have been obtained on roughly ~500×500 sized images using their default number of proposals, as reported in their respective papers. Because some studies ([9, 20, 21]) do not provide details on processing time, we make an estimation based on details of their approach (denoted by *). In the figure, we also include information on some well known fully-supervised R-CNN approaches [6, 7, 12, 14, 15, 29] for reference. As can be seen, the VGG-16 based SPAM-CAM performs about 10-15 times faster than all other weakly supervised approaches. In fact, even a CPU-only implementation of our approach roughly performs in the same speed range as other TitanX/K40 GPU based implementations. Additionally, we are able to match the speeds of existing fully supervised single-shot methods like [6, 7, 29].

2.5 Conclusion

In this chapter, a convolutional-only single-stage architecture extension based on Class Activation Maps (CAM) is demonstrated for the task of weakly supervised object localisation in real-time without the use of region proposals. Concurrently, a novel global Spatial Pyramid Averaged Max (SPAM) pooling technique is introduced that is used for training such a CAM augmented deep network for localising objects in an image using only weak image-level (presence/absence) supervision. This SPAM pooling layer is shown to posses a suitable flow of backpropagating gradients during weakly supervised training. This forms a good middle ground between the strong single-point gradient flow of global max pooling and the equal spread gradient flow of global average pooling for ascertaining the extent of the object in the image. Due to this, the proposed approach requires only a single forward pass through the network, and utilises a fast-backprojection algorithm to provide bounding boxes for an object without any costly region proposal steps, resulting in real-time inference. The method is validated on the PASCAL VOC datasets and is shown to produce good accuracy, while being able to perform inference at 35fps, which is 10–15 times faster than all other related frameworks.

References

[1] A. Gudi, N. Van Rosmalen, M. Loog, and J. Van Gemert, *Object extent pooling for weakly supervised single-shot localization,* in *British Machine Vision Conference 2017, BMVC 2017*, British Machine Vision Conference 2017, BMVC 2017 (BMVA Press, 2017) 28th British Machine Vision Conference, BMVC 2017 ; Conference date: 04-09-2017 through 07-09-2017.

[2] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, *Weakly supervised object localization with progressive domain adaptation,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 3512–3520.

[3] H. Bilen and A. Vedaldi, *Weakly supervised deep detection networks,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 2846–2854.

[4] Y. Li, K. He, J. Sun, *et al.*, *R-fcn: Object detection via region-based fully convolutional networks,* in *Advances in Neural Information Processing Systems* (2016) pp. 379–387.

[5] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, *Feature pyramid networks for object detection,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *Ssd: Single shot multibox detector,* in *European Conference on Computer Vision* (2016) pp. 21–37.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 779–788.

[8] A. J. Bency, H. Kwon, H. Lee, S. Karthikeyan, and B. Manjunath, *Weakly supervised localization using deep feature maps,* in *European Conference on Computer Vision* (Springer, 2016) pp. 714–731.

[9] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, *Is object localization for free? - weakly-supervised learning with convolutional neural networks,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015) pp. 685–694.

[10] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, *Learning deep features for discriminative localization,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 2921–2929.

[11] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, *Selective search for object recognition,* International Journal of Computer Vision **104**, 154 (2013).

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation,* in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2014) pp. 580–587.

[13] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition,* in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2016) pp. 770–778.

[14] R. Girshick, *Fast r-cnn,* in *Proceedings of the IEEE International Conference on Computer Vision* (2015) pp. 1440–1448.

[15] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks,* in *Advances in neural information processing systems* (2015) pp. 91–99.

[16] B. Alexe, T. Deselaers, and V. Ferrari, *Measuring the objectness of image windows,* IEEE transactions on Pattern Analysis and Machine Intelligence **34**, 2189 (2012).

[17] I. Endres and D. Hoiem, *Category-independent object proposals with diverse ranking,* IEEE transactions on Pattern Analysis and Machine Intelligence **36**, 222 (2014).

[18] P. O. Pinheiro, R. Collobert, and P. Dollar, *Learning to segment object candidates,* in *Advances in Neural Information Processing Systems* (2015) pp. 1990–1998.

[19] J. Long, E. Shelhamer, and T. Darrell, *Fully convolutional networks for semantic segmentation,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015) pp. 3431–3440.

[20] C. Wang, W. Ren, K. Huang, and T. Tan, *Weakly supervised object localization with latent category learning,* in *European Conference on Computer Vision* (2014) pp. 431–445.

[21] R. G. Cinbis, J. Verbeek, and C. Schmid, *Weakly supervised object localization with multi-fold multiple instance learning,* IEEE transactions on Pattern Analysis and Machine Intelligence **39**, 189 (2017).

[22] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, *Solving the multiple instance problem with axis-parallel rectangles,* Artificial intelligence **89**, 31 (1997).

[23] T. Hofmann, *Probabilistic latent semantic analysis,* in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* (Morgan Kaufmann Publishers Inc., 1999) pp. 289–296.

[24] M. Lin, Q. Chen, and S. Yan, *Network in network,* in *International Conference on Learning Representations* (2014).

[25] Y.-L. Boureau, J. Ponce, and Y. LeCun, *A theoretical analysis of feature pooling in visual recognition,* in *Proceedings of the 27th International Conference on Machine Learning* (2010) pp. 111–118.

[26] Y. LeCun *et al.*, *Generalization and network design strategies,* Connectionism in Perspective , 143 (1989).

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition,* Proceedings of the IEEE **86**, 2278 (1998).

[28] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition,* in *International Conference on Learning Representations* (2015).

[29] J. Redmon and A. Farhadi, *Yolo9000: better, faster, stronger,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017).

3

Input and Data Efficiency in Webcam Gaze Tracking

Efficiency and ease of use are essential for practical applications of camera based eye/gaze-tracking. Gaze tracking involves estimating where a person is looking on a screen based on face images from a computer-facing camera. In this chapter, we investigate two complementary forms of efficiency in gaze tracking: 1. The computational efficiency of the system which is dominated by the inference speed of a CNN predicting gaze-vectors; 2. The usability efficiency which is determined by the tediousness of the mandatory calibration of the gaze-vector to a computer screen. To do so, we evaluate the computational speed/accuracy trade-off for the CNN and the calibration effort/accuracy trade-off for screen calibration. For the CNN, we evaluate the full face, two-eyes, and single eye input. For screen calibration, we measure the number of calibration points needed and evaluate three types of calibration: 1. pure geometry, 2. pure machine learning, and 3. hybrid geometric regression. Results suggest that a single eye input and geometric regression calibration achieve the best trade-off.

This chapter is based on *Gudi, A., li, X., & van Gemert, J. (2020). Efficiency in Real-time Webcam Gaze Tracking. In A. Bartoli, & A. Fusiello (Eds.), Computer Vision – ECCV 2020 Workshops: Proceedings (1 ed., pp. 529 - 543). (Part of the Lecture Notes in Computer Science book series (LNCS, volume 12535) Also part of the Image Processing, Computer Vision, Pattern Recognition, and Graphics book sub series (LNIP, volume 12535); Vol. 12535). Springer.* https://doi.org/10.1007/978-3-030-66415-2_34 [1].

3.1 Introduction

In a typical computer-facing scenario, the task of gaze-tracking involves estimating where a subject's gaze is pointing based on images of the subject captured via the webcam. This is commonly in the form of a gaze vector, which determines the pitch and yaw of the gaze with respect to the camera [2]. A more complete form of gaze tracking further extends this by also computing at which specific point the subject is looking at on a screen in front of the subject [3, 4]. This is achieved by estimating the position of the said screen w.r.t. the camera (a.k.a. screen calibration), which is not precisely known beforehand. We present a study of some core choices in the design of gaze estimation methods in combination with screen calibration techniques (see Figure 3.1), leaning towards an efficient real-time camera-to-screen gaze-tracking system.

Computational efficiency: Input size

Deep networks, and CNNs in particular, improved accuracy in gaze estimation where CNN inference speed is to a large extent determined by the input image size. The input size for gaze estimation can vary beyond just the image of the eye(s) [2, 5], but also include the whole eye region [4], the whole face, and even the full camera image [3]. Yet, the larger the input image, the slower the inference speed. We study the impact of various input types and sizes with varying amounts of facial contextual information to determine their speed/accuracy trade-off.

Usability efficiency: Manual effort in screen calibration

Most work focus on gaze vectors estimation [6–10]. However, predicting the *gaze-point*, point on a screen in front of the subject where he/she is looking, is a more intuitive and directly useful result for gaze tracking based applications, especially in a computer-facing/human-computer interaction scenario. If the relative locations and pose of the camera w.r.t to the screen were exactly known, projecting the gaze-vector to a point on screen would be straightforward. However, this transformation is typically not known in real-world scenarios, and hence must also be implicitly or explicitly estimated through an additional calibration step. This calibration step needs to be repeated for every setup. Unlike gaze-vector prediction, you cannot have a "pre-trained" screen calibration method. In practice, every-time a new eye-tracking session starts, the first step would be to ask the user to look at and annotate predefined points for calibration. Therefore, obtaining calibration data is a major usability bottleneck since it requires cooperation of the user every time, which in practice varies. Here, we study usability efficiency as a trade-off between the number of calibration points and accuracy.

We consider three types of calibration. Geometry based modelling methods have the advantage that maximum expert/geometrical prior knowledge can be embedded into the system. On the other hand, such mathematical models are rigid and based on strong assumptions, which may not always hold. In contrast, calibration methods based on machine learning require no prior domain knowledge and limited hand-crafted modelling. However, they may be more data-dependent in order to learn the underlying geometry. In this chapter, we evaluate the efficiency trade-off of various calibration techniques including a hybrid approach between machine learning regression and geometric modelling.

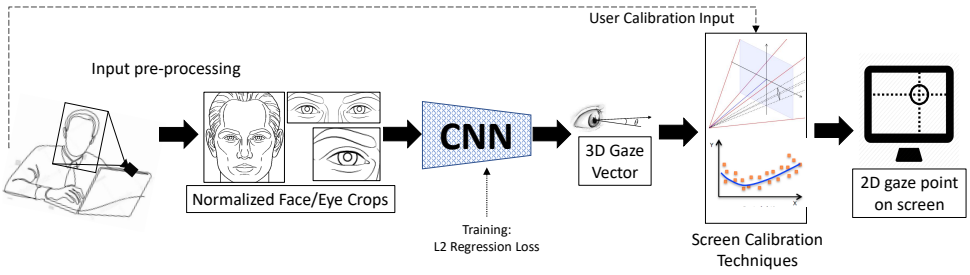


Figure 3.1: An overview illustration of our camera-to-screen gaze tracking pipeline under study. (Left to right) Images captured by the webcam are first pre-processed to create a normalized image of face and eyes. These images are used for training a convolutional neural network (using L2 loss) to predict the 3D gaze vector. With the cooperation of the user, the predicted gaze vectors can finally be projected on to the screen where he/she is looking using proposed screen calibration techniques.

Contributions

We have the following three contributions:

(i) We evaluate computational system efficiency by studying the balance of gains from context-rich inputs vs their drawbacks. We study their individual impact on the system's accuracy w.r.t. their computational load to determine their efficiency and help practitioners find the right trade-off.

(ii) We demonstrate three practical screen calibration techniques that can be use to convert the predicted gaze-vectors to points-on-screen, thereby performing the task of complete camera-to-screen gaze tracking.

(iii) We evaluate the usability efficiency of these calibration methods to determine how well they utilize expensive user-assisted calibration data. This topic has received little attention in literature, and we present one of the first reports on explicit webcam-based screen calibration methods.

3.2 Related Work

Existing methods for gaze tracking can be roughly categorized into model-based and appearance-based methods. The former [11, 12] generates a geometric model for eye to predict gaze, while the latter [13] makes a direct mapping between the raw pixels and the gaze angles. Appearance driven methods have generally surpassed classical model-based methods for gaze estimation.

Appearance-based CNN gaze-tracking

As deep learning methods have shown their potentials in many areas, some appearance-based CNN networks are shown to work effectively for the task of gaze prediction.

Zhang *et al.* [2, 14] proposed the first deep learning model for appearance-based gaze prediction. Park *et al.* [5] proposed a combined hourglass [15] and DenseNet [16] network to

take advantage of auxiliary supervision based on the gaze-map, which is two 2D projection binary mask of the iris and eyeball. Cheng *et al.* [17] introduced ARE-Net, which is divided into two smaller modules: one is to find directions from each eye individually, and the other is to estimate the reliability of each eye. Deng and Zhu *et al.* [18] define two CNNs to generate head and gaze angles respectively, which are aggregated by a geometrically constrained transform layer. Ranjan *et al.* [19] clustered the head pose into different groups and used a branching structure for different groups. Chen *et al.* [7] proposed Dilated-Nets to extract high level features by adding dilated convolution. We build upon these foundations where we evaluate the speed vs accuracy trade-off in a real-time setting. The image input size has a huge effect on processing speed, and we control the input image size by varying eye/face context.

The seminal work of Zhang *et al.* [2, 14] utilized minimal context by only using the grayscale eye image and head pose as input. Krafka *et al.* [3] presented a more context-dependent multi-model CNN to extract information from two single eye images, face image and face grid (a binary mask of the face area in an image). To investigate how the different face region contributes to the gaze prediction, a full-face appearance-based CNN with spatial weights was proposed [4]. Here, we investigate the contribution of context in the real-time setting by explicitly focusing on the speed/accuracy trade-off.

A GPU based real-time gaze tracking method was presented in [6]. This was implemented in a model ensemble fashion, taking two eye patches and head pose vector as input, and achieved good performance on several datasets [2, 6, 20] for person-independent gaze estimation. In addition, [7, 8] have included some results about the improvements that can be obtained from different inputs. In our work, we perform an ablation study and add the dimension of computation load of each input type. Our insight in the cost vs benefit trade-off may help design efficient gaze tracking software that can run real-time beyond expensive GPUs, on regular CPUs which have wider potential in real world applications.

Screen calibration: Estimating point-of-gaze

In a classical geometry-based model, projecting any gaze-vector to a point on a screen requires a fully-calibrated system. This includes knowing the screen position and pose in the camera coordinate system. Using a mirror-based calibration technique [21], the corresponding position of camera and screen can be attained. This method needs to be re-applied for different computer and camera setting, which is non-trivial and time-consuming. During human-computer interactions, information like mouse clicks may also provide useful information for screen calibration [22]. This is, however, strongly based on the assumption that people are always looking at the mouse cursor during the click.

Several machine learning models are free of rigid geometric modelling while showing good performance. Methods like second order polynomial regression [23] and Gaussian process regression [24] have been applied to predict gaze more universally. WebGazer [22] trains regression models to map pupil positions and eye features to 2D screen locations directly without any explicit 3D geometry. As deep learning features have shown robustness in different areas, other inputs can be mixed with CNN-based features for implicit calibration, as done in [3, 4]. CNN features from the eyes and face are used as inputs to a support vector regressor to directly estimate gaze-point coordinates without an explicit calibration step. These methods take advantage of being free of rigid modelling and show good

performance. On the other hand, training directly on CNN features makes this calibration technique non-modular since it is designed specific to a particular gaze-prediction CNN. In our work, we evaluate data-efficiency for modular screen calibration techniques that convert gaze-vectors to gaze-points based on geometric modelling, machine learning, and a mix of geometry and regression. We explicitly focus on real world efficiency which for calibration is not determined by processing speed, but measured in how many annotations are required to obtain reasonable accuracy.

3.3 Setup

The pipeline contains three parts, as illustrated in Figure 3.1:

1. Input pre-processing by finding and normalizing the facial images;
2. A CNN that takes these facial images as input to predict the gaze vector;
3. Screen calibration and converting gaze-vectors to points on the screen.

3.3.1 Input Pre-processing

The input to the system is obtained from facial images of subjects. Through a face finding and facial landmark detection algorithm [25], the face and its key parts are localized in the image. Following the procedure described by Sugano *et al.* [20], the detected 2D landmarks are fitted onto a 3D model of the face. This way, the facial landmarks are roughly localized in the 3D camera coordinate space. By comparing the 3D face model and 2D landmarks, the head rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$, and the 3D eye locations $\mathbf{e}$ are obtained in 3D camera coordinate space. A standardized view of the face is now obtained by defining a fixed distance d between the eye centres and the camera centre and using a scale matrix $\mathbf{S} = \text{diag}(1, 1, \frac{d}{\|\mathbf{e}\|})$. The obtained conversion matrix $\mathbf{M} = \mathbf{S} \cdot \mathbf{R}$ is used to apply perspective warping to obtain a normalized image without roll (in-plane rotation). For training, the corresponding ground truth vector $\mathbf{g}$ is similarly transformed: $\mathbf{M} \cdot \mathbf{g}$.

3.3.2 CNN Prediction of Gaze Vectors

We use a VGG16 [26] network architecture with BatchNorm [27] to predict the pitch and yaw angles of the gaze vector with respect to the camera from the normalized pre-processed images.

Training Following the prior work in[2], the network was pre-trained on ImageNet [28]. For all the experiments conducted in this work, we set the following hyperparameters for the training of the network for gaze-vector prediction:

(i) Adam optimizer with default settings [29];

(ii) a validation error based stopping criteria with a patience of 5 epochs;

(iii) learning rate of 10^{-5}, decaying by 0.1 if validation error plateaus;

(iv) simple data augmentation with mirroring and gaussian noise ($\sigma = 0.01$).

Inference This trained deep neural network can now make prediction of the gaze vector. The predicted gaze-vector (in the form of pitch and yaw angles) are with respect to the 'virtual' camera corresponding to the normalized images. The predicted virtual gaze vectors can be transformed back to the actual gaze vector with respect to the real camera using the transformation parameters obtained during image pre-processing. These vectors can then be projected onto a point on the screen after screen calibration.

3.3.3 Screen Calibration: Gaze Vectors to Gaze Points

To project the predicted 3D gaze vectors (in the camera coordinate space) to 2D gaze-points on a screen, the position of the screen with respect to the camera must be known which is difficult to obtain in real world settings. The aim of screen calibration is to estimate this geometric relation between the camera and the screen coordinate systems such that the predicted gaze vectors in camera coordinates are calibrated to gaze-points in screen coordinates. Because we focus on the task of eye-tracking in a computer-facing scenario, we can simplify the setup by making some assumptions based on typical webcam-monitor placement (such as for built-in laptop webcams or external webcams mounted on monitors):

(i) the roll and yaw angles between the camera and the screen are 0°,

(ii) the intrinsic camera matrix parameters are known, and

(iii) the 3D location of the eye is roughly known w.r.t the camera (estimated by the eye landmarks in the face modelling step in camera coordinate space).

With these assumptions in place, we can design user-aided calibration techniques where the user cooperates by looking at predefined positions on the screen.

3.4 Screen Calibration Methods

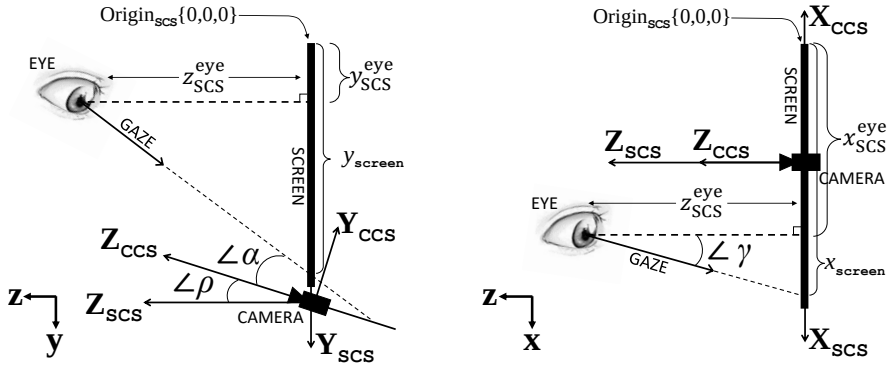
As calibration is tedious and needs to be performed multiple times, we evaluate efficiency in terms of how much manual effort is required for three calibration versions:

1. calibration by geometry;
2. calibration by machine learning;
3. calibration by a hybrid: geometry and regression.

3.4.1 Geometry-based Calibration

To perfectly project a gaze-vector w.r.t the camera to a point on a screen, we are essentially required to determine the transformation parameters between the camera coordinate system (CCS) and the screen coordinate system (SCS). With our assumptions about roll and yaw in place, this transformation can be expressed by the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{T}$ between the camera and the screen:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\rho) & -\sin(\rho) \\ 0 & \sin(\rho) & \cos(\rho) \end{bmatrix} \quad \& \quad \mathbf{T} = \begin{bmatrix} \Delta x & \Delta y & \Delta z \end{bmatrix}^T, \tag{3.1}$$



(a) $y-z$ plane with pitch angles (b) $x-z$ plane with yaw angle

Figure 3.2: A illustration of the geometric setup between the eye and the screen in the screen coordinate space. $\{\mathbf{X},\mathbf{Y},\mathbf{Z}\}_{CCS}$ and $\{\mathbf{X},\mathbf{Y},\mathbf{Z}\}_{SCS}$ represent the directions of the $\{x,y,z\}$-axes of the camera and screen coordinate systems respectively; Origin$_{SCS}$ represents the origin of the screen coordinate system.

where ρ denotes the vertical pitch angle (about the x-axis; along the y-axis) between the camera and the screen norm, and $\Delta x, \Delta y, \Delta z$ represent the translational displacement between the camera and the screen. An illustration of the geometric setup is shown in Figure 3.2 (3.2a and 3.2b).

Step 1 The first step is to estimate the location of eye in the screen coordinate system. This can be done with the aid of the user, who is asked to sit at a pre-set distance z from the screen and look perpendicular at the screen plane (such that the angle between the gaze vector and the screen plane becomes 90°). He is then instructed to mark the point of gaze on the screen, denoted by $\{x,y\}$. In this situation, these marked screen coordinates would directly correspond to the x and y coordinates of the eye in the screen coordinate space. Thus, the eye location can be determined as: $\mathbf{e}_{SCS} = \{x^{eye}_{SCS}, y^{eye}_{SCS}, z^{eye}_{SCS}\} = \{x,y,z\}$.

During this time, the rough 3D location of the eye in the camera coordinate system is also obtained (from the eye landmarks of the face modelling step) and represented as $\mathbf{e}_{CCS}$. With this pair of corresponding eye locations obtained, the translation vector T can be expressed by:

$$\mathbf{e}_{SCS} = \mathbf{R}\cdot\mathbf{e}_{CCS} + \mathbf{T} \quad \Longrightarrow \quad \mathbf{T} = \mathbf{e}_{SCS} - \mathbf{R}\cdot\mathbf{e}_{CCS}. \tag{3.2}$$

Step 2 Next, without (significantly) changing the head/eye position, the user is asked to look at a different pre-determined point on the screen $\{x_{screen}, y_{screen}\}$.

During this time, the gaze estimation system is used to obtain the gaze direction vector in the camera coordinate system:

$$\mathbf{g}_{CCS} = \begin{bmatrix} x^{gaze}_{CCS} & y^{gaze}_{CCS} & z^{gaze}_{CCS} \end{bmatrix}^T. \tag{3.3}$$

This is a normalized direction vector whose values denote a point on a unit sphere. Both the pitch α (about the x-axis) and the yaw γ (about the y-axis) angles of the gaze w.r.t the

camera can be re-obtained from this gaze direction vector:

$$\alpha = \arctan 2(-y_{CCS}^{gaze}, z_{CCS}^{gaze}) \quad \& \quad \gamma = \arctan 2(x_{CCS}^{gaze}, z_{CCS}^{gaze}). \tag{3.4}$$

Once α is determined, we can calculate the camera pitch angle ρ between the camera and the screen:

$$\rho = \arctan(\frac{-y_{SCS}^{eye} + y_{screen}}{z_{SCS}^{eye}}) - \alpha. \tag{3.5}$$

Using this in Equation 3.1, the rotation matrix $\mathbf{R}$ can be fully determined. This known $\mathbf{R}$ can now be plugged into Equation 3.2 to also determine the translation vector $\mathbf{T}$. This procedure can be repeated for multiple calibration points in order to obtain a more robust aggregate estimate of the transformation parameters.

Step 3 Once calibration is complete, any new eye location $\hat{\mathbf{e}}_{CCS}$ can be converted to the screen coordinate space:

$$\hat{\mathbf{e}}_{SCS} = [\hat{x}_{SCS}^{eye}, \hat{y}_{SCS}^{eye}, \hat{z}_{SCS}^{eye}] = \mathbf{R} \cdot \hat{\mathbf{e}}_{CCS} + \mathbf{T}. \tag{3.6}$$

Using the associated new gaze angles $\hat{\alpha}$ and $\hat{\gamma}$, the point of gaze on the screen can be obtained:

$$\hat{x}_{screen} = \hat{z}_{SCS}^{eye} \cdot tan(\hat{\gamma}) + \hat{x}_{SCS}^{eye} \quad \& \quad \hat{y}_{screen} = \hat{z}_{SCS}^{eye} \cdot tan(\hat{\alpha} + \rho) + \hat{y}_{SCS}^{eye}, \tag{3.7}$$

3.4.2 Machine Learning (ML)-based Calibration

Since the task of gaze vector to gaze point calibration requires learning the mapping between two sets of coordinates, this can be treated as a regression problem. In our implementation, we use a linear ridge regression model for this task for it's ability to avoid overfitting when training samples are scarce. The input to this calibration model includes the predicted gaze-vector angles and the 3D location of the eye, all in the camera coordinate system. The outputs are the 2D coordinates of the gaze-point on the screen in the screen coordinate system.

During calibration, the user is asked to look at a number of predefined points on the screen (such that they span the full region of the screen) while their gaze and eye locations are estimated and recorded for each of these points. These calibration samples are then used to train the model. Given enough training/calibration points, this model is expected to implicitly learn the mapping between the two coordinate systems.

3.4.3 'Hybrid' Geometric Regression Calibration

To combine the benefits of geometry based prior knowledge with ML based regression, a hybrid geometric regression technique can be derived where machine learning is used to infer the required geometric transformation parameters.

As before, we assume the roll and yaw angles between the camera and the screen are 0°. The only unknown between the pose of the camera w.r.t the screen is the pitch angle ρ. The rotation and translation matrices are the same as given by Equation 3.1, and the formulations of gaze pitch and yaw angles α and γ stay the same as defined by Equation 3.4.

Again, during calibration the user is asked to look at a number of varied predefined points on the screen while their gaze directions and eye locations are recorded. These data samples are then used to jointly minimize the reprojection errors (squared Euclidean distance) to learn the required transformation parameters $\rho, \Delta x, \Delta y, \Delta z$:

$$\underset{\rho,\Delta x,\Delta y,\Delta z}{\arg\min} \sum_{i=1}^{N} \left((x^i_{\texttt{point}} - \hat{x}^i_{\texttt{screen}})^2 + (y^i_{\texttt{point}} - \hat{y}^i_{\texttt{screen}})^2 \right), \tag{3.8}$$

where N is the number of training/calibration points; $\{x_{\texttt{point}}, y_{\texttt{point}}\}$ denote the ground truth screen points, while $\{\hat{x}_{\texttt{screen}}, \hat{y}_{\texttt{screen}}\}$ are the predicted gaze points on screen as estimated using Equation 3.7.
We solve this minimization problem by differential evolution [30].

3.5 Experiments and Results

3.5.1 Datasets

We perform all our experiments on two publicly available gaze-tracking datasets:

MPIIFaceGaze [4] This dataset is an extended version of MPIIGaze [2] with available human face region. It contains 37,667 images from 15 different participants. The images have variations in illumination, personal appearance, head pose and camera-screen settings. The ground truth gaze target on the screen is given as a 3D point in the camera coordinate system.

EYEDIAP [31] This dataset provides 94 video clips recorded in different environments from 16 participants. It has two kinds of gaze targets: screen points and 3D floating targets. It also has two types of head movement conditions: static and moving head poses. For our experiments, we choose the screen point target with both the static and moving head poses, which contains 218,812 images.

3.5.2 Exp 1: Speed/Accuracy Trade-off for Varying Input Sizes

Setup

A gaze vector can be predicted based on various input image sizes, as illustrated in Figure 3.3:

- Full face image: The largest and most informative input;
- Two eyes: Medium sized and informative;
- Single eye: Minimal information and smallest size.

To assess the performance gains of different input sizes vs their computational loads and accuracy, we setup an experiment where we vary the input training and testing data to the neural network while keeping all other settings fixed. We then measure the accuracy of the system and compute their individual inference-time computational loads.

For this experiment, we individually train our deep network on each of the multiple types and sizes of the pre-processed inputs shown in Figure 3.3. In order to obtain a reliable error metric, we perform 5-fold cross-validation training. This experiment is repeated for both the MPIIFaceGaze and EYEDIAP dataset.

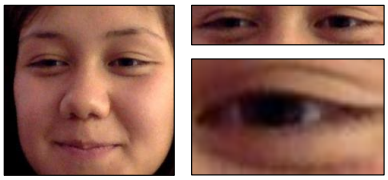


Figure 3.3: Examples of three input types used in the experiments: (left) face crop, sized 224×224 and 112×112; (right top) two eyes region crop, sized 180×60 and 90×30; (right bottom) single eye crop, sized 60×32 and 30×18.

Results

The results of this experiment can be seen in Figure 3.4. As expected, we observed that the lowest error rates are obtained by the largest size of input data with the maximum amount of context: the full face image. We also observe that using this input type results in the highest amount of computation load.

As we reduce the input sizes, the accuracy only marginally degrades while the computation load gets cut down severely. In fact, even if we simply use a crop of the eye region or just the crop of a single eye, we obtain accuracies comparable to that from full face input albeit with a fraction of the computation.

3.5.3 Exp 2: Screen Calibration Usability/Accuracy Trade-off

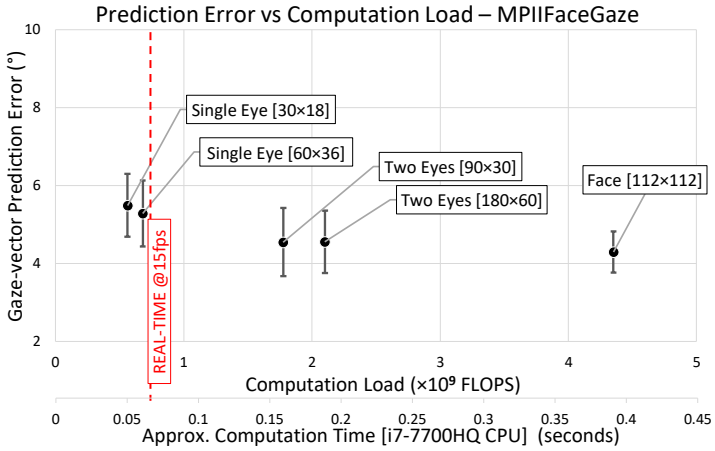
Setup

To evaluate the three screen calibration techniques proposed, we train and test them individually using calibration data samples from MPIIFaceGaze and EYEDIAP. This data for screen calibration consists of pairs of gaze-vectors and their corresponding ground truth screen points. Using this, calibration methods are trained to predict the 2D screen points from the 3D gaze-vectors. We evaluate on noise-free ground truth gaze-vectors and on realistic predicted gaze-vectors (using 30×18 eye crop as input) so as to assess the accuracy of the complete camera-to-screen eye-tracking pipeline. As training data, we obtain calibration data pairs of gaze-vectors and points such that they are spread out evenly over the screen area. This is done by dividing the screen in an evenly-spaced grid and extracting the same number of points from each grid region.

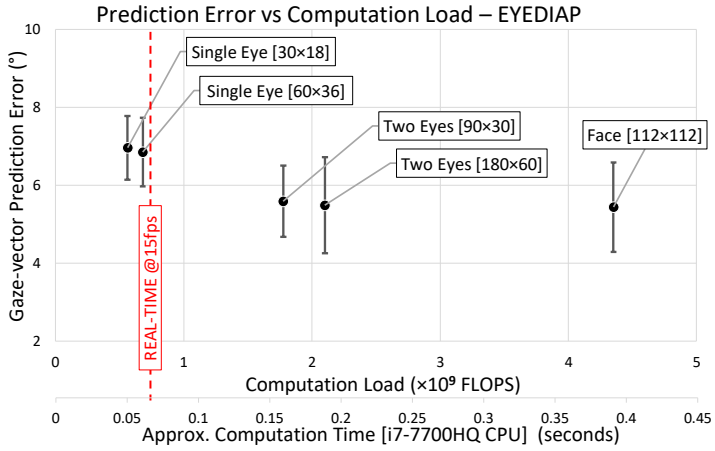
Results

The results of these experiments can be seen in Table 3.1 for a fixed calibration training set size of 100 samples. For the 'theoretical' task of predicting gaze-points from noise-free ground truth gaze-vectors, we see in Table 3.1a that the hybrid geometric regression method outperforms others. We see that the gap in performance is smaller when head poses are static, while the hybrid method does better for moving head poses. This suggests that for the simplest evaluation on static head poses with noise-free gaze-vectors, all methods perform well; however, as movement is introduced, the limitations of the purely geometric method and the advantage of hybrid method becomes clearer.

When calibration is performed on actual gaze-vectors predicted by the system, overall accuracy deteriorates by one to three orders of magnitude. Comparing the methods, the hybrid geometric regression method also does well compared to others in most conditions, as seen in Table 3.1b. We observe that the purely geometric method actually copes better than the ML based method when head poses are static. However, it's performance severely degrades with moving head poses. Also, the ML method is able to marginally outperform



(a) MPIIFaceGaze



(b) EYEDIAP

Figure 3.4: Scatter plots of the performance of a VGG16 based gaze tracking network trained on different input types vs their computation load/time in FLOPS/seconds. The error bars represent the standard deviation of the errors (5-fold cross-validation). While the computation cost of these inputs vastly vary, they all perform in roughly the same range of accuracy. The red dashed line represents approximate real-time computation at 15 fps on an *Intel i7* CPU.

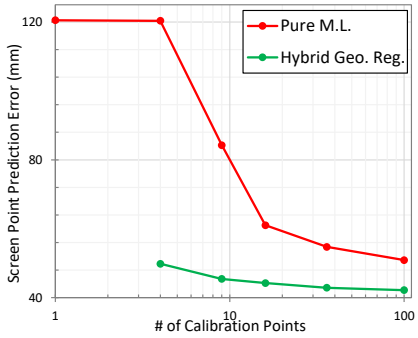
Ground Truth Gaze Vector Dataset	Screen Calibration Method [Prediction Error in *mm*] Pure Geometric	Pure M.L.	Hybrid Geo. Reg.
MPIIFaceGaze	N/A	9.27	**1.23**
EYEDIAP [Static]	5.98	2.73	**2.35**
EYEDIAP [Moving]	22.45	8.55	**2.39**

(a) Groundtruth gaze-vector calibration

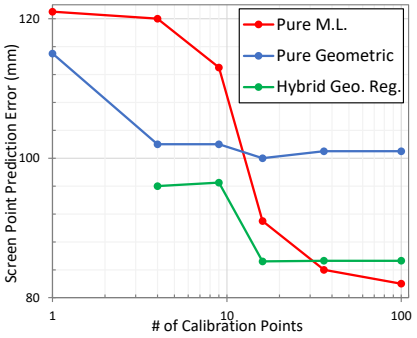
Predicted Gaze Vector Dataset	Screen Calibration Method [Prediction Error in *mm*] Pure Geometric	Pure M.L.	Hybrid Geo. Reg.
MPIIFaceGaze	N/A	50.92	**42.19**
EYEDIAP [Static]	67.72	80.63	**61.6**
EYEDIAP [Moving]	101.53	**82.7**	86.37

(b) Predicted gaze-vector calibration

Table 3.1: Performance of calibration methods (trained with 100 samples) on different datasets and conditions expressed in gaze-point prediction errors (in *mm*). Hybrid geometric regression technique significantly outperforms both purely geometric and purely machine learning (M.L.) based calibration methods in most conditions. Legend: [Static] denotes static head poses, [Moving] denotes moving head poses.



(a) MPIIFaceGaze



(b) EYEDIAP (Moving)

Figure 3.5: Learning curves of the calibration techniques on MPIIFaceGaze and EYEDIAP dataset (log scale). The purely geometric method performs better than ML method when calibration data is scarce, but does not improve further when more data is available. The ML method improves greatly when calibration data becomes abundant. The hybrid geometric regression method performs the best over a wide range of calibration data points used.

the hybrid method on moving head poses. This is likely because given sufficient training samples, the ML method is able to learn features from the input that the other—more rigid—methods cannot do. Note that only the EYEDIAP dataset results are reported for the purely geometric technique, since this method can only be trained on static head poses and MPIIFaceGaze does not have any static head poses (the geometric method can still be tested on the moving head poses of EYEDIAP).

To assess the efficiency of these calibration methods, we must ascertain the least amount of calibration samples required with which satisfactory performance can still be attained. This can be assessed by observing the learning curves of the calibration methods, where the prediction errors of the methods are plotted against the number of calibration/training points used. This is shown in Figure 3.5.

The hybrid method is able to outperform both the other methods even when a low number of calibration points are available. An interesting observation seen in Figure 3.5b is that the purely geometric method actually performs better than the ML method when

the number of calibration points is low (9). This can be due to the rigid and pre-defined nature of the geometric model which has prior knowledge strongly imparted into it. On the other hand, the ML model requires more data points to learn the underlying geometry from scratch. This is also seen in the results: as the number of points increase, the ML model's performance improves while the geometric model stagnates. Overall, the lower error rate of the hybrid model over a broad range of used calibration points affirms its strengths over the overtly rigid geometric model and the purely data-driven ML approach.

3.6 Discussion

The experiments related to input types and sizes produce some insightful and promising results. The comparison between them with respect to their performance vs their computational load indicate that the heavier processing of larger inputs with more contextual information is not worth the performance gain they produce. Roughly the same accuracies can be obtained by a system that relies only on eye image crops. In contrast, the gap in the computational load between these two input types is a factor of 20. This supports our idea that for an objective measurement task like gaze-vector prediction, the value of context is limited. These results can help in guiding the design of eye tracking systems meant for real-time applications where efficiency is key.

Outputs in the form of gaze-vectors are not always readily useful in a computer-facing scenario: they need to be projected onto the screen to actually determine where the person is looking. This area has received little attention in literature, and our experiments provide some insight. Our comparison of three calibration techniques show that a hybrid geometric regression method gives the overall best performance over a wide range of available calibration data points. Our results show that purely geometric modelling works better when calibration points are very few, while a purely ML method outperforms it when more points become available. However, a hybrid model offers a robust trade-off between them.

3.7 Conclusion

In this work, we explored the value of visual context in input for the task of gaze tracking from camera images. Our study gives an overview of the accuracy different types and sizes of inputs can achieve, in relation to the amount of computation their analysis requires. The results strongly showed that the improvement obtained from large input sizes with rich contextual information is limited while their computational load is prohibitively high. Additionally, we explored three screen calibration techniques that project gaze-vectors onto screens without knowing the exact transformations, achieved with the cooperation of the user. We showed that in most cases, a hybrid geometric regression method outperforms a purely geometric or machine learning based calibration while generally requiring less calibration data points and thus being more efficient.

References

[1] A. Gudi, X. li, and J. van Gemert, *Efficiency in real-time webcam gaze tracking,* in *Computer Vision – ECCV 2020 Workshops*, Part of the Lecture Notes in Computer Science book series (LNCS, volume 12535); also part of the Image Processing, Computer Vision, Pattern Recognition, and Graphics book sub series (LNIP, volume 12535), edited by A. Bartoli and A. Fusiello (Springer, 2020) pp. 529 – 543, European Conference on Computer Vision (ECCV) 2020 Workshop on Eye Gaze in AR, VR, and in the Wild (OpenEyes), ECCVW 2020; OpenEyes 2020; Conference date: 23-08-2020 Through 28-08-2020.

[2] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, *Mpiigaze: Real-world dataset and deep appearance-based gaze estimation,* IEEE Transactions on Pattern Analysis and Machine Intelligence **41**, 162 (2017).

[3] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, *Eye tracking for everyone,* in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[4] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, *It's written all over your face: Full-face appearance-based gaze estimation,* in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2017) pp. 2299–2308.

[5] S. Park, A. Spurr, and O. Hilliges, *Deep pictorial gaze estimation,* in *European conference on computer vision* (2018).

[6] T. Fischer, H. Jin Chang, and Y. Demiris, *Rt-gene: Real-time eye gaze estimation in natural environments,* in *Proceedings of the European Conference on Computer Vision* (2018).

[7] Z. Chen and B. E. Shi, *Appearance-based gaze estimation using dilated-convolutions,* in *Proceedings of the Asian Conference on Computer Vision* (2018) pp. 309–324.

[8] C. Palmero, J. Selva, M. A. Bagheri, and S. Escalera, *Recurrent CNN for 3d gaze estimation using appearance and shape cues,* in *British Machine Vision Conference (BMVC)* (2018).

[9] S. Park, S. D. Mello, P. Molchanov, U. Iqbal, O. Hilliges, and J. Kautz, *Few-shot adaptive gaze estimation,* in *Proceedings of the IEEE International Conference on Computer Vision* (2019) pp. 9368–9377.

[10] Y. Yu, G. Liu, and J.-M. Odobez, *Improving few-shot user-specific gaze adaptation via gaze redirection synthesis,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019) pp. 11937–11946.

[11] E. Wood and A. Bulling, *Eyetab: Model-based gaze estimation on unmodified tablet computers,* in *Proceedings of the Symposium on Eye Tracking Research and Applications* (ACM, 2014) pp. 207–210.

[12] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, *A 3d morphable model of the eye region,* in *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Posters* (2016) pp. 35–36.

[13] K.-H. Tan, D. J. Kriegman, and N. Ahuja, *Appearance-based eye gaze estimation,* in *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.* (IEEE, 2002) pp. 191–195.

[14] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, *Appearance-based gaze estimation in the wild,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015) pp. 4511–4520.

[15] A. Newell, K. Yang, and J. Deng, *Stacked hourglass networks for human pose estimation,* in *European conference on computer vision* (Springer, 2016) pp. 483–499.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, *Densely connected convolutional networks,* in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2017) pp. 4700–4708.

[17] Y. Cheng, F. Lu, and X. Zhang, *Appearance-based gaze estimation via evaluation-guided asymmetric regression,* in *Proceedings of The European Conference on Computer Vision* (2018).

[18] H. Deng and W. Zhu, *Monocular free-head 3d gaze tracking with deep learning and geometry constraints,* in *Proceedings of the International Conference on Computer Vision* (2017) pp. 3162–3171.

[19] R. Ranjan, S. De Mello, and J. Kautz, *Light-weight head pose invariant gaze tracking,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2018) pp. 2156–2164.

[20] Y. Sugano, Y. Matsushita, and Y. Sato, *Learning-by-synthesis for appearance-based 3d gaze estimation,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014) pp. 1821–1828.

[21] R. Rodrigues, J. P. Barreto, and U. Nunes, *Camera pose estimation using images of planar mirror reflections,* in *European Conference on Computer Vision* (2010).

[22] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, *Webgazer: Scalable webcam eye tracking using user interactions,* in *Proceedings of the International Joint Conference on Artificial Intelligence* (2016) pp. 3839–3845.

[23] P. Kasprowski, K. Harezlak, and M. Stasch, *Guidelines for the eye tracker calibration using points of regard,* in *Information Technologies in Biomedicine, Volume 4* (Springer International Publishing, 2014) pp. 225–236.

[24] S. Tripathi and B. Guenter, *A statistical approach to continuous self-calibrating eye gaze tracking for head-mounted virtual reality systems,* in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (IEEE, 2017) pp. 862–870.

[25] T. Baltrušaitis, P. Robinson, and L.-P. Morency, *Continuous conditional neural fields for structured regression,* in *Proceedings of the European Conference on Computer Vision* (2014) pp. 593–608.

[26] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition,* in *International Conference on Learning Representations* (2015).

[27] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift,* in *International Conference on Machine Learning* (2015) pp. 448–456.

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge,* International Journal of Computer Vision (IJCV) , 211 (2015).

[29] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization,* in *Proceedings of the International Conference on Learning Representations* (2015).

[30] R. Storn and K. Price, *Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,* Journal of Global Optimization , 341 (1997).

[31] K. A. Funes Mora, F. Monay, and J.-M. Odobez, *Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras,* in *Proceedings of the ACM Symposium on Eye Tracking Research and Applications* (2014).

4

Exploiting Spatial Context for Anomaly Detection and Feature Learning

Mean squared error (MSE) is one of the most widely used metrics to expression differences between multi-dimensional entities, including images. However, MSE is not locally sensitive as it does not take into account the spatial arrangement of the (pixel) differences, which matters for structured data types like images. Such spatial arrangements carry information about the source of the differences; therefore, an error function that also incorporates the location of errors can lead to a more meaningful distance measure. We introduce Proximally Sensitive Error (PSE), through which we suggest that a regional emphasis in the error measure can 'highlight' semantic differences between images over syntactic/random deviations. We demonstrate that this emphasis can be leveraged upon for the task of anomaly/occlusion detection. We further explore its utility as a loss function to help a model focus on learning representations of semantic objects instead of minimizing syntactic reconstruction noise.

This chapter is based on *Gudi, A., Büttner, F., & van Gemert, J. (2019). Proximally Sensitive Error for Anomaly Detection and Feature Learning. Extended abstract, ICT.OPEN, Hilversum, 2019. ArXiv:2206.00506.* `https://arxiv.org/abs/2206.00506` [1]

4.1 Introduction

Since its introduction by Eukleídēs [2] in ~300BC, Euclidean/L2 distance has widely been used to express distances between points. The simplicity, robustness, and mathematical convenience of this metric makes it a popular choice as a similarity/distance metric in computer vision and machine learning in the form of mean squared error (MSE) [3–5], where it finds wide use for all domains of data types including images [6, 7]. In this work, we present an alternate to MSE for structured data types (images) in the form of a proximally sensitive error function.

MSE is well suited for measuring differences between scalars which are single-dimensional, or vectors which are multi-dimensional. Such data types have no information in the spatial arrangement of values/elements within, i.e., there are no patterns in the way their elements are located (they are only required to be consistent). For example, a feature vector describing an object's important characteristics. For such non-structured data types, the relative locations (indexes) of the differences does not matter. On the other hand, for structured data types like images, relative locations of the differences do matter because information is also encoded in the the underlying spatial arrangement of values. For example, pixels composing a visual image of an object. However, this is not considered by the mean squared error, as illustrated with an example in Figure 4.1 (column 3). In this study, we attempt to address this drawback of MSE by introducing a locally sensitive metric.

An error function that incorporates spatial location of errors can lead to a more meaningful error metric for images, since the meaning of the content of an image relies heavily on the location of the pixel values that represent semantic or meaningful objects. Based on the observation that adjacent pixels often form regions of semantic meaning (in other words: objects in the image) while sparse spread-out errors are caused by random or syntactic noise, we hypothesise that an error function that emphasizes regions with high concentrations of pixel-wise reconstruction error forms a better metric (see example in Figure 4.1, column 4). Towards this end, we introduce Proximally Squared Error (PSE), that implements this spatial dependence via Gaussian convolutions.

In the field of computer vision, one of the utilities of such a location sensitive error metric can be in tasks involving image reconstruction. Typically for such tasks, the difference between a reconstructed image and the original image is computed to iteratively improve the reconstruction [6, 8] and/or make a downstream classification [9, 10]. To empirically evaluate our hypothesis, we examine proximally sensitive error against mean squared error for the tasks of image anomaly detection and unsupervised pre-training, both involving image reconstruction.

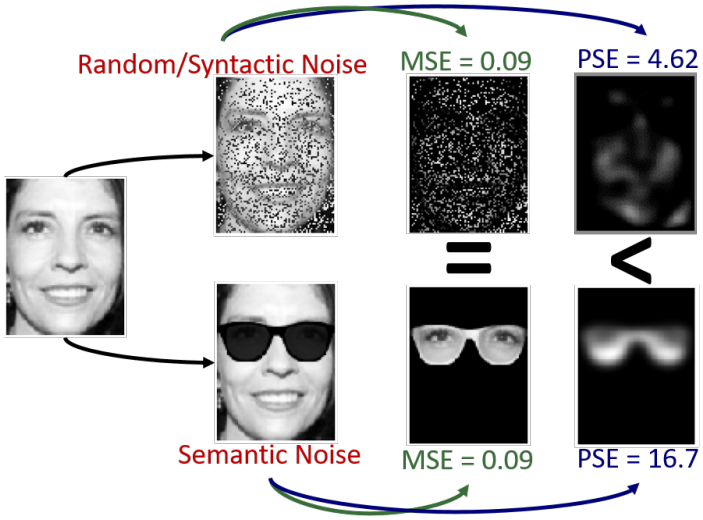


Figure 4.1: Comparison PSE vs MSE: A face image (left) is subtracted by two variants (col 2): one with sunglasses (bottom), and one with some random noise addition (top). MSE between the original image and the two variants is equal (col 3). However, PSE of the image with sunglasses is much higher than with random noise (col 4). This illustrates that MSE cannot distinguish between syntactic/random and semantic/meaningful errors, while PSE can.

Contributions This work has the following contributions:

(i) We present proximally sensitive error (PSE), a novel locally sensitive error function that also takes into account the relative locations of differences between such structured data types.

(ii) We examine and provide insights into the applicability of PSE versus MSE for image reconstruction powered tasks of anomaly detection and unsupervised pre-training.

4.2 Related Work

Distance metrics for images The straight-forward pixel-wise L2/Euclidian distance, equivalent to the mean squared error (MSE), is a popular choice for expressing differences between images [4–7]. MSE has been used in the image reconstruction terms of loss functions for training several neural network approaches such as sparse autoencoders [4], variational autoencoders [5], convolutional autoencoders [7], and generative adversarial networks [6]. In this work, we propose proximally sensitive error (PSE) as an alternative to MSE to express differences between images.

MSE does not match well with visually perceived differences between images by humans [11, 12]. To counter this, Wang *et al.* [13] introduced the structured similarity metric (SSIM), an distance metric specifically designed for assessment of image/video quality loss due to compression. SSIM is able to look at neighbouring pixels within a pre-defined window. Concurrently, Li *et al.* [14] discovered heuristics from a large dataset to design the dynamic partial distance function (DPF) that better represents perceptual

similarity. In our work, we propose a simplified distance function for structured data like images by incorporating regional sensitivity.

Image anomaly detection Anomaly detection in images can be divided into two broad classes (among others) [15, 16]: direct inference/classification of anomalies [17–19]; and reconstruction and comparison based anomaly detection (typically un/semi-supervised) [8, 20, 21]. Bergmann *et al.* [22] examined SSIM as a distance metric for autoencoder reconstruction anomaly detection, which yielded results similar to L2. In our work, we focus on a PCA reconstruction based anomaly detection setup where we examine the applicability of PSE as a distance metric.

Unsupervised pre-training Unsupervised pre-training was discovered to provide a superior alternative to fully stochastic parameter initialization of deep neural networks [23–25]. The use of end-to-end pre-training on image data has been widely studied due to its potential benefits [9, 10], typically using the pixel-wise L2/MSE reconstruction loss for optimization. In this chapter, we explore PSE as a location-sensitive image reconstruction loss for unsupervised pre-training with images, and examine if this can lead to better feature learning.

4.3 Method

For two-dimensional images, the residuals between two images is simply the differences between individual pixel values of the two images. This can be expressed mathematically as:

$$\mathcal{R}_{i,j} = \hat{Y}_{i,j} - Y_{i,j} \qquad \forall \;\; i \in M, \;\; j \in N \tag{4.1}$$

where $Y_{i,j}$, $\hat{Y}_{i,j}$ represents the i^{th} and j^{th} pixel in images Y and $\hat{Y}$ of size $M \times N$.

The Mean Squared Error (MSE) between these two images is essentially the average of the squares of the residuals between the two images. This is expressed as:

$$\text{MSE} = \frac{1}{MN} \sum_{i,j=1}^{M,N} \left(\mathcal{R}_{i,j}\right)^2 \tag{4.2}$$

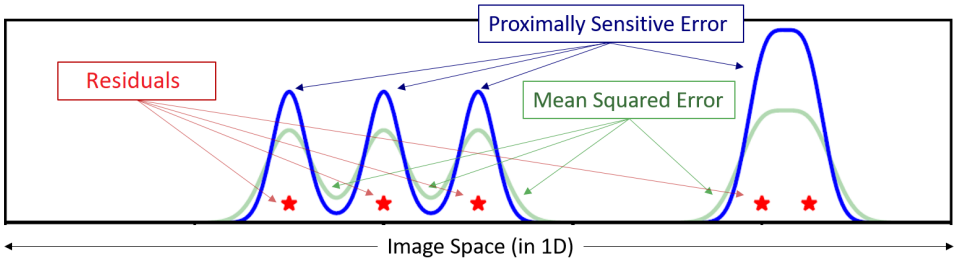


Figure 4.2: Illustration MSE vs PSE: MSE magnitude in the areas with more spread-out errors is the same as areas with concentrated grouping of residuals. In contract, PSE error is higher for closely grouped areas.

As can be seen, the squared error values do not take the neighbourhood of individual errors into consideration. Thus, residuals that are grouped tightly and residuals that are spread-out contribute equally towards the mean. This is illustrated in Figure 4.2.

To counter this, we propose a Proximally Sensitive Error (PSE) function:

$$\text{PSE} = \frac{1}{MN} \sum_{i,j=1}^{M,N} \left([\mathcal{R} * k(\sigma)]_{i,j} \right)^2 \tag{4.3}$$

where $*$ denotes the convolution operation and $k(\sigma)$ denotes a Gaussian kernel with σ as its standard deviation, given by:

$$k(\sigma)_{x,y} = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad \forall \; x, y \in 2\sigma \tag{4.4}$$

Through the convolution operation, every error value incorporates the error value in its local neighbourhood (defined by σ). This way, PSE is able to give a higher importance to residuals grouped close together versus residuals sparsely spread-out. This is also visualized in Figure 4.2.

Image reconstruction based anomaly detection

Assuming closely grouped errors denote the more meaningful/semantic differences between images (our hypothesis), we can leverage upon PSE's ability to highlight them to perform anomaly detection. The pipeline for anomaly detection based on image reconstruction involves the following steps, and illustrated in Figure 4.3.

- A principle component analysis (PCA) model [26], which is capable of image reconstruction, is trained on purely non-anomalous images of a particular object/class.
- The input image is passed through this PCA model and its reconstructed image is obtained. Due to the model's training, the non-anomalous parts of the input image are reconstructed correctly, but the model fails to reconstruct any anomalies in the image that it hasn't seen during its training. This effect can further be enhanced by reducing the number of PCA components used to reconstruct the image.
- Next, the pixelwise PSE (with a set σ parameter) between the original and reconstructed image is computed. This pixelwise PSE 'image' can essentially be seen as a heatmap of differences between the two images.
- Finally, the maximum PSE value in this computed heatmap is chosen denoting the area with the highest difference, and this value provides and estimate of anomalousness in the image.

The σ parameter of the PSE function and the number of PCA components used for image reconstruction can be optimized/learnt (e.g., via grid search) based on examples of anomalous and non-anomalous images available in the training set.

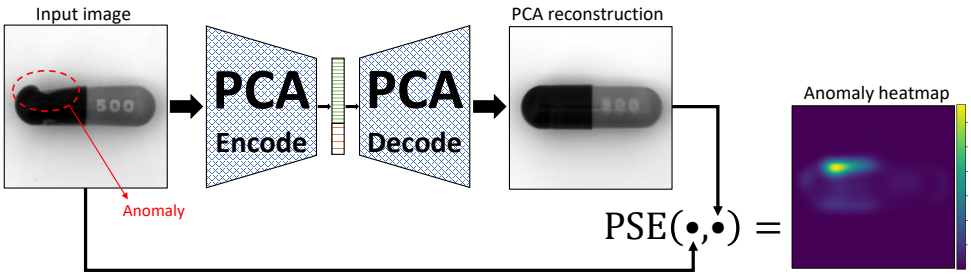


Figure 4.3: Illustration of the anomaly detection pipeline using proximally sensitive error (PSE). A PCA model trained on non-anomalous images is unable to reconstruct anomalies present in the input (the crushed part of the capsule in this example). PSE can be used to highlight this difference, as seen in the produced heatmap.

Feature learning via unsupervised pre-training

If the closely-grouped errors assigned a higher value by PSE are indeed more semantic in nature (our hypothesis), PSE can essentially perform as a more meaningful loss function for unsupervised image reconstructing. This is because the model can be made to focus more on minimizing the reconstruction of the closely grouped semantic errors over the spread-out syntactic errors during training, leading to better feature learning.

Such unsupervised image reconstruct can be used as a pre-training step for a downstream task such as classification using a autoencoder-style neural network. The steps involved in this setup are as follows given a semi-supervised dataset of unlabelled and labelled samples:

- First, an autoencoder is trained end-to-end for image reconstruction with PSE as the loss function using the large set of unlabelled samples.
- Next, the encoder part of the autoencoder is appended with classification layer(s) (e.g. fully connected layer with softmax activation) while the decoder is discarded. This new model is now trained on the smaller set of labelled samples.

Image classification can now be performed with this trained encoder-classifier model.

4.4 Experiments and Results

4.4.1 Anomaly/Occlusion Detection

We focus on the task of few-shot anomaly detection in these experiments.

Setup The task of anomaly detection consists of classifying if a particular object in a given input image has an abnormality. For example, picture of a bottle with a crack in it can be classified as anomalous since normal bottles do not have cracks. Under the few-shot learning regime for anomaly detection, the vast majority of available training samples belong to the non-anomalous class, and only a handful of samples contain anomalies.

Datasets We perform this set of experiments on two publicly available datasets: The MVTecAD dataset [22] for industrial anomaly detection, and the AR Face dataset [27] for

facial occlusion detection. The MVTecAD dataset consists of ~5300 images of 15 categories of object classes, with ~4100 non-anomalous samples and ~1200 images with multiple types of anomalies. The AR Face dataset consists of 2600 frontal images of faces with varying illumination and facial expressions, with 1200 of them containing occlusions caused by sunglasses and scarf. Images from both datasets are resized and converted to 128×128 grayscale. Examples from these datasets can be seen in Figure 4.5 (first rows).

Industrial Anomaly Detection

The results of 5-shot industrial anomaly detection on the MVTecAD dataset are shown in Figure 4.4a in terms of average precision. Only 5 labelled samples per anomaly class are used for optimizing the σ hyper-parameter of the PSE function and the number of PCA components used for image reconstruction.

As can be seen, the use of PSE attains a higher average precision (0.73±0.2) than MSE (0.66±0.3) on average. Better performance is obtained by PSE w.r.t MSE on 11 of the 15 object categories in the dataset. In few categories like *toothbrush* and *bottle*, the performance gap between PSE and MSE is marginal while the performance is very high. This is likely due to the straight-forward appearance of anomalies in the images, thereby making the task trivial for both PSE and MSE.

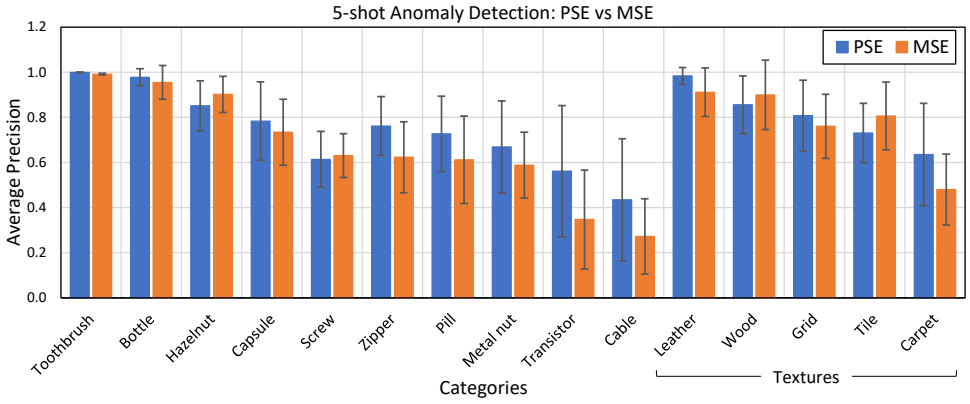
On the other hand, images from two of the categories where PSE performer poorer than MSE (*hazelnut* and *screw*) are not pose-normalized and contain rotation. This results in a very high variation in terms of the object's pose, which a simple PCA model is unable to model and reconstruct. For images of the *tile* category, some anomalies are in the form of transparent occlusions which leads to unreliable performance. Lastly, anomaly detection performance for *cable* is the lowest for both MSE and PSE. This is because most anomalies in this category are in the colour space (e.g., differently coloured wires are swapped), and this information is lost due to the grayscale conversion. These observations can be seen in the examples shown in Figure 4.5.

Facial Occlusion Detection

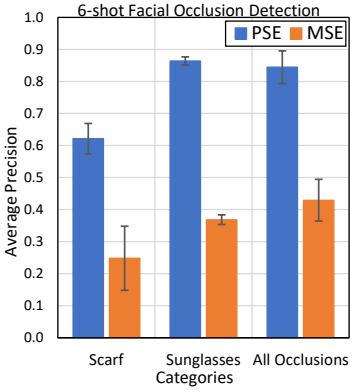
Figure 4.4b summarises and compares the results of 5-shot facial occlusion detection on the AR Face dataset using PSE and MSE. Similar to the previous experiment, 5 labelled samples per class are used to determine the model hyper-parameters (σ and PCA components).

It can be seen that PSE significantly outperforms MSE: PSE obtains an average precision of 0.84 ± 0.05 while MSE scores 0.43 ± 0.07 over all classes. Both PSE and MSE perform better with detecting sunglasses as compared to scarf, likely due to the scarf occluding the lower part of the face that contains more variation due to facial hair and expressions.

Overall, PSE based occlusion detection on faces from the AR dataset works a lot better than anomaly detection on the MVTecAD dataset. This can be attributed to the pose-normalisation of face crops in AR and the much higher number of training samples available, as compared to that of the individual object categories of MVTecAD.

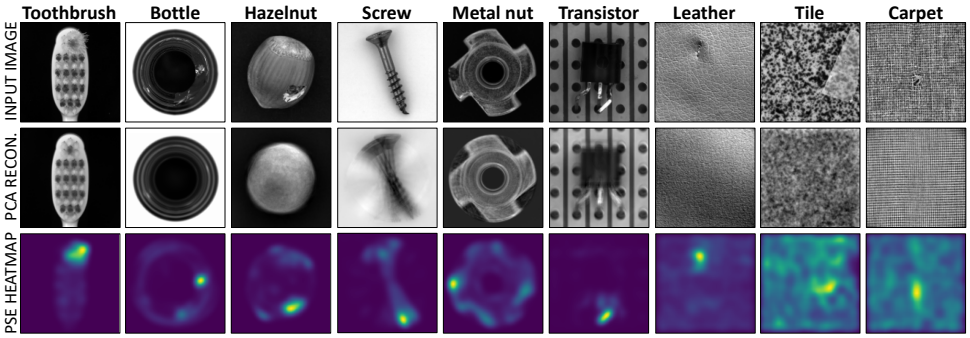


(a) MVTecAD Industrial Dataset

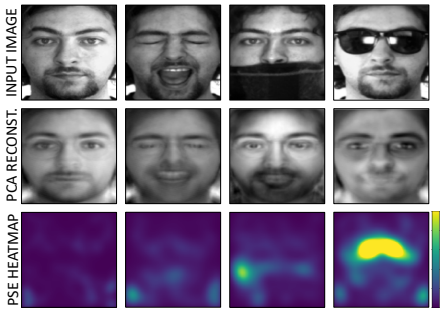


(b) AR Face Dataset

Figure 4.4: Results of 5-shot anomaly/occlusion detection on the MVTecAD industrial dataset ((a)) and AR face dataset ((b)). PSE performs better than MSE in almost all categories. Poor performance in some categories are due lack of pose normalization (*screw*, *hazelnut*), transparent occlusions (*tile*), loss of colour information (*cable*), and higher variation in lower face (*scarf*).



(a) MVTecAD Industrial Dataset



(b) AR Face Dataset

Figure 4.5: Examples of 5-shot anomaly/occlusion detection on the MVTecAD industrial dataset ((a)) and AR face dataset ((b)). The PCA is unable to reconstruct (middle rows) the anomaly in the input image (top rows). Using this, PSE essentially provides a heatmap of the anomaly (bottom rows). Reconstructions of *screw* and *metal nut* are especially poor due lack of pose-normalization in the images.

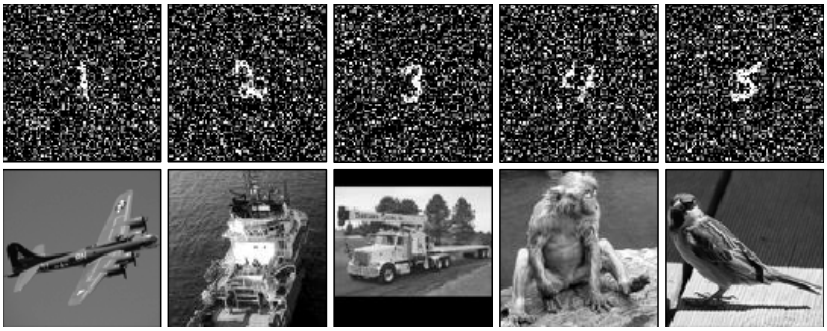


Figure 4.6: Example images from the modified MNISTX (top row) and STL-10 datasets (bottom row). The depicted classes are (left to right): 1, 2, 3, 4, 5 from MNIST; and airplane, ship, truck, monkey, bird from STL-10.

4.4.2 Unsupervised Pre-training

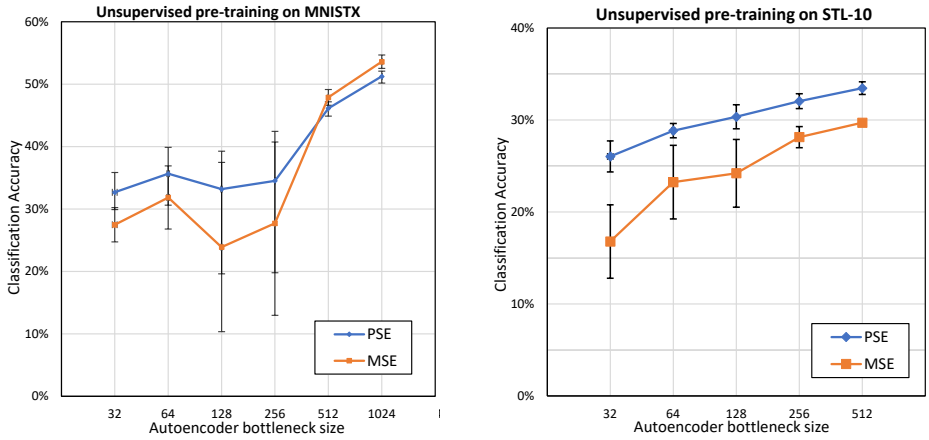
In these experiments, we evaluate the use of PSE vs MSE as a loss function for the task of unsupervised pre-training.

Datasets Two classification datasets are used in this set of experiments: a modified version of MNIST handwritten digits dataset [28] (hereby called MNISTX) composed of 70,000 labelled samples (10 classes); and a grayscale version of the STL-10 image classification dataset [29] containing 100,000 unlabelled and 500 labelled images (10 classes). Examples from both datasets are shown in Figure 4.6. The MNIST dataset has been modified by padding the original image such that its size is tripled (84×84 pixels) and adding salt-and-pepper noise to the image. The STL-10 dataset images are 96×96 pixels and converted to grayscale.

Setup We consider a simple autoencoder neural network architecture composed on an input layer, a fully connected hidden layer with ReLU that serves as the bottleneck, and a fully connected output layer with a sigmoid activation function. Another fully-connected softmax layer serves as the classification layer whose dimensions match the number of classes in the dataset, i.e., 10. The dimensions of the input and output layers match the number of pixels in the input images (i.e., 84×84 for MNISTX and 96×96 for STL-10). The σ parameter of the PSE loss function is set to 0.5 (determined empirically).

Computational Efficiency

The resulting accuracy obtained by the model using PSE and MSE with respect to the autoencoder bottleneck size (i.e., dimension of the hidden layer) can be seen in Figure 4.7. For both datasets, it can be seen that using PSE results in higher accuracy than MSE on average per bottleneck size. This means PSE can achieve the same results as MSE in spite of using smaller models. In the STL-10 dataset, the accuracy gap between PSE and MSE is largest when the autoencoder's bottleneck is the smallest, and this gap reduces gradually as the bottleneck size increases. On the MNISTX dataset, such an observation is not as clear (standard deviations overlap), however the accuracy gap does become smaller/inverts for large bottleneck sizes.



(a) MNISTX (b) STL-10

Figure 4.7: Results of unsupervised pre-training using PSE and MSE on MNISTX ((a)) and STL-10 ((b)) datasets in terms of the autoencoder bottleneck size (error bars represent standard deviations). PSE performs better than MSE for models with smaller bottlenecks, but this gap reduces for higher complexity models.

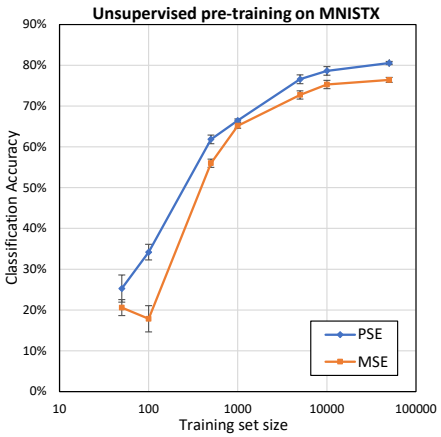
These observations can be attributed to the the fact that a model with fewer parameters benefits the most from PSE's ability to focus the learning on minimizing semantic errors, thereby leading to more computationally efficient feature learning. On the other hand, the higher learning capacity of larger models might enable them to learn meaningful features in spite of no explicit boosting of semantic errors over syntactic ones in MSE.

Data Efficiency

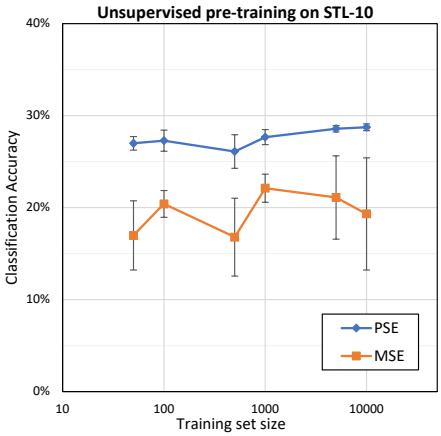
Figure 4.8 shows the accuracy obtained by using PSE and MSE for varying amounts of unsupervised pre-training data used. As can be seen, PSE consistently outperforms MSE as a loss function on both datasets over all sizes of the unsupervised pre-training set. This suggests fewer data is required by PSE to achieve the same results as MSE.

On MNISTX, the accuracy gap between PSE and MSE is larger for lower training set sizes, and this gap appears to close in when more training data is introduced. This could suggest that PSE's focus on more meaningful errors helps the model learn more efficiently from the data, and this effect is magnified when the available training data is limited.

On the STL-10 dataset, such an observation is not apparent. However, PSE seems to produce more consistent results than MSE, both across and within different training set sizes (the standard deviation of accuracies is smaller). This could potentially be caused by the strong attention on the more semantic errors in PSE overcoming the effect of stochasticity in the model's parameter initialization that could have lead to learning distractions.



(a) MNISTX



(b) STL-10

Figure 4.8: Results of unsupervised pre-training using PSE and MSE on MNISTX and STL-10 datasets in terms of the size of the pre-training set (error bars represent training sets). Overall, PSE performs better than MSE over all sizes of training data used. PSE results are also more consistent, evident by the smaller standard deviation error bars.

4.5 Discussion

Experiments performed on the application of proximally sensitive error (PSE) for few-shot anomaly detection appeared to show promising results. PSE was able to outperform mean squared error (MSE) in detecting anomalies on a large majority of object categories. This potentially supports our suggestion that closely grouped differences are primarily semantic in nature and PSE is able to boost them in comparison with spread-out differences, which are largely syntactic, i.e., caused by random noise.

In these experiments, the parameter defining the amount of spread of errors (i.e., the σ of the Gaussian kernel) did require to be estimated/optimized per object category. This suggests that for different types of objects, the spatial definition of semantic objects vs syntactic noise is different. Also, inferring/learning this parameter automatically (along with the number of PCA components used) can improve the applicability of the proposed anomaly detection pipeline from few-shot towards zero-shot detection.

Experiments on the use of PSE as a loss function for unsupervised pre-training suggested that PSE can lead to better feature learning than MSE under constrained conditions: when the model's computational capacity is limited, or when the availability of training data is low. This can make PSE a promising choice for training models for low-powered computational devices and novel image tasks. However, the results lack certainty and a wider range of experiments on different types of datasets and more complex model architectures can shed further light on the generalizability of these results.

References

[1] A. Gudi, F. Büttner, and J. van Gemert, *Proximally sensitive error for anomaly detection and feature learning,* in *ArXiv:2206.00506; Extended Abstract, ICT.OPEN, Hilversum* (2019).

[2] L. C. Bruno, *Math and mathematicians : the history of math discoveries around the world* (1999).

[3] C.-P. Lee and C.-J. Lin, *A study on l2-loss (squared hinge-loss) multiclass svm,* Neural computation **25**, 1302 (2013).

[4] A. Ng *et al.*, *Sparse autoencoder,* CS294A Lecture notes **72**, 1 (2011).

[5] D. P. Kingma and M. Welling, *Auto-encoding variational bayes,* arXiv preprint arXiv:1312.6114 (2013).

[6] X. Mao, Q. Li, H. Xie, R. Y. Lau, and Z. Wang, *Multi-class generative adversarial networks with the l2 loss function,* arXiv preprint arXiv:1611.04076 **5**, 1057 (2016).

[7] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, *Deep features learning for medical image analysis with convolutional autoencoder neural network,* IEEE Transactions on Big Data (2017).

[8] C. Zhou and R. C. Paffenroth, *Anomaly detection with robust deep autoencoders,* in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (2017) pp. 665–674.

[9] S. Wiehman, S. Kroon, and H. De Villiers, *Unsupervised pre-training for fully convolutional neural networks,* in *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)* (IEEE, 2016) pp. 1–6.

[10] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin, *Unsupervised pre-training of image features on non-curated data,* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019) pp. 2959–2968.

[11] B. Girod, *Psychovisual aspects of image processing: What's wrong with mean squared error?* in *Proceedings of the seventh workshop on multidimensional signal processing* (IEEE, 1991) pp. P–2.

[12] P. Sinha and R. Russell, *A perceptually based comparison of image similarity metrics,* Perception **40**, 1269 (2011).

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: from error visibility to structural similarity,* IEEE transactions on image processing **13**, 600 (2004).

[14] B. Li, E. Chang, and Y. Wu, *Discovery of a perceptual distance function for measuring image similarity,* Multimedia systems **8**, 512 (2003).

[15] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, *Deep learning for anomaly detection: A review,* ACM Computing Surveys (CSUR) **54**, 1 (2021).

[16] T. Ehret, A. Davy, J.-M. Morel, and M. Delbracio, *Image anomalies: A review and synthesis of detection methods,* Journal of Mathematical Imaging and Vision **61**, 710 (2019).

[17] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, and K.-R. Müller, *Explainable deep one-class classification,* arXiv preprint arXiv:2007.01760 (2020).

[18] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, *Attention guided anomaly localization in images,* in *European Conference on Computer Vision* (Springer, 2020) pp. 485–503.

[19] I. Golan and R. El-Yaniv, *Deep anomaly detection using geometric transformations,* arXiv preprint arXiv:1805.10917 (2018).

[20] S. Hawkins, H. He, G. Williams, and R. Baxter, *Outlier detection using replicator neural networks,* in *International Conference on Data Warehousing and Knowledge Discovery* (Springer, 2002) pp. 170–180.

[21] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, *Outlier detection with autoencoder ensembles,* in *Proceedings of the 2017 SIAM international conference on data mining* (SIAM, 2017) pp. 90–98.

[22] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, *The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection,* International Journal of Computer Vision **129**, 1038 (2021).

[23] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, *Greedy layer-wise training of deep networks,* in *Advances in neural information processing systems* (2007) pp. 153–160.

[24] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, *The difficulty of training deep architectures and the effect of unsupervised pre-training,* in *Artificial Intelligence and Statistics* (PMLR, 2009) pp. 153–160.

[25] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, *Why does unsupervised pre-training help deep learning?* in *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (JMLR Workshop and Conference Proceedings, 2010) pp. 201–208.

[26] H. Hotelling, *Analysis of a complex of statistical variables into principal components.* Journal of educational psychology **24**, 417 (1933).

[27] A. Martinez and R. Benavente, *The ar face database,* (1998).

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition,* Proceedings of the IEEE **86**, 2278 (1998).

[29] A. Coates, A. Ng, and H. Lee, *An analysis of single-layer networks in unsupervised feature learning,* (2011).

5

Prior Knowledge driven Efficient Vital Signs Estimation from Faces

Remote photo-plethysmography (rPPG) uses a camera to estimate a person's heart rate (HR). Similar to how heart rate can provide useful information about a person's vital signs, insights about the underlying physio/psychological conditions can be obtained from heart rate variability (HRV). HRV is a measure of the fine fluctuations in the intervals between heart beats. However, this measure requires temporally locating heart beats with a high degree of precision. We introduce a refined and efficient real-time rPPG pipeline with novel filtering and motion suppression that not only estimates heart rates, but also extracts the pulse waveform to time heart beats and measure heart rate variability. This unsupervised method requires no rPPG specific training and is able to operate in real-time. We also introduce a new multi-modal video dataset, VicarPPG 2, specifically designed to evaluate rPPG algorithms on HR and HRV estimation. We validate and study our method under various conditions on a comprehensive range of public and self-recorded datasets, showing state-of-the-art results and providing useful insights into some unique aspects. Lastly, we make available CleanerPPG, a collection of human-verified ground truth peak/heart-beat annotations for existing rPPG datasets. These verified annotations should make future evaluations and benchmarking of rPPG algorithms more accurate, standardized and fair.

This chapter is based on:

Gudi, A., Bittner, M., & van Gemert, J. (2020). Real-Time Webcam Heart-Rate and Variability Estimation with Clean Ground Truth for Evaluation. Applied Sciences, 10(23), 1-24. [8630]. https://doi.org/10.3390/app10238630 [1]; and

Gudi, A., Bittner, M., Lochmans, R., & van Gemert, J. (2019). Efficient real-time camera based estimation of heart rate and its variability. In Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019 (pp. 1570-1579). [9022193]. https://doi.org/10.1109/ICCVW.2019.00196 [2].

5.1 Introduction

Human vital signs like heart rate, blood oxygen saturation and related physiological measures can be measured using a technique called photo-plethysmography (PPG). This technique involves optically monitoring light absorption in tissues that are associated with blood volume changes. Typically, this is done via a contact sensor attached to the skin surface [3]. Such contact sensors can detect the underlying vital signs quite reliably owing to their proximity to the subject, and therefore have applications in critical areas like patient monitoring. However, the ability to obtain such measurements remotely via a camera/webcam, albeit less accurately, can enable applications outside the medical domain (e.g. affective computing, human-computer interaction), where contact sensors are not feasible. Remote Photo-plethysmography (rPPG) detects the blood volume pulse remotely by tracking changes in the skin reflectance as observed by a camera [4, 5]. In this chapter, we present a novel framework for extracting heart rate (HR) and heart rate variability (HRV) from the face. This work is an extension of the work done in [2].

Vital signs from videos The process of rPPG essentially involves two steps: detecting and tracking the skin colour changes of the subject, and analysing this signal to compute measures like heart rate, heart rate variability and respiration rate. Recent advances in computer video, signal processing, and machine learning have improved the performances of rPPG techniques significantly [4]. Current state-of-the-art methods are able to leverage image processing by supervised deep neural networks to robustly select skin pixels within an image and perform HR estimation [6, 7]. However, this reliance upon heavy machine learning (ML) processes has two primary drawbacks: (i) it necessitates rPPG specific fully supervised training of the ML model, thereby requiring collection of large training sets; (ii) complex models can require significant computation time on CPUs and thus can potentially add a bottleneck in the pipeline and limit real-time utility. Since rPPG analysis is based on a signal processing task, the use of an end-to-end trainable system with no domain knowledge leaves room for improvement in efficiency (e.g., we *know* that pulse signal is embedded in average skin colour changes [5, 8, 9], but the machine learning system has to *learn* this). We introduce an efficient unsupervised rPPG pipeline that performs the full rPPG analysis in real-time. This method achieves state-of-the-art results without needing any rPPG related training. This is achieved via extracting regions of interest robustly by 3D face modelling, and explicitly tracking and reducing the influence of head movement to filter the signal.

Heart rate variability While heart rate is a useful output from a PPG/rPPG analysis, finer analysis of the obtained blood volume pulse (BVP) signal can yield further useful measures. One such measure is heart rate variability (HRV): an estimate of the variations in the time-intervals between individual heart beats. This measure has utility in providing insights into the physiological and psychological state of a person (stress levels, anxiety, etc.). While traditionally this measure is obtained based on observation over hours [10], short and ultra-short duration ($\leq$ 5 mins) HRV are also being studied [11]. Our experiments focus on obtaining ultra-short HRV measure as a proof-of-concept/technology demonstrator for longer duration applications.

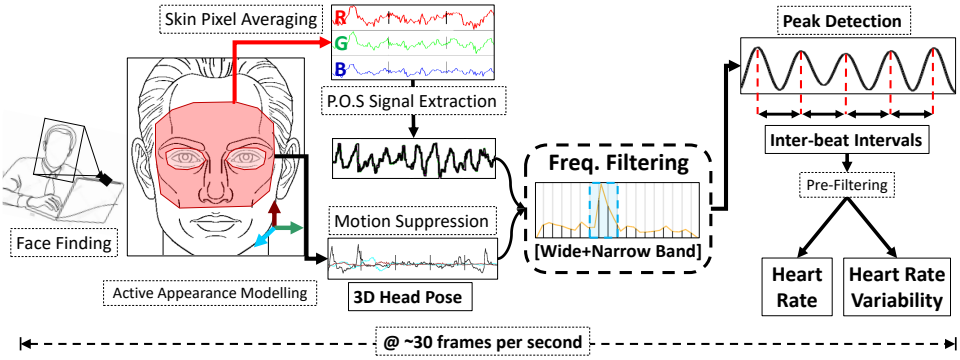


Figure 5.1: An overview of the proposed heart rate and heart rate variability estimation pipeline (left to right). The face in captured webcam images are detected and modelled to track the skin pixels in region of interest. A single 1-D signal is extracted from the spatially averaged values of these pixels over time. In parallel, 3-D head movements are tracked and used to suppress motion noise. An FFT based wide and narrow band filtering process produces a clean pulse waveform from which peaks are detected. The inter-beat intervals obtained from these peaks are then filtered and used to compute heart rate and heart rate variability. The full analysis can be performed in real time on a CPU.

The computation of heart rate variability depends on the time-variations between heart beats, and it therefore requires temporally locating the heart beats with a high degree of accuracy. Unlike HR estimation, where errors in opposite directions average out, HRV analysis is sensitive to even small artefacts and all errors add up to strongly distort the final measurement. Thus, estimating HRV is a challenging task for rPPG and this has received relatively little focus in literature. Our method extracts a clean BVP signal from the input via a two-step wide and narrow band frequency filter to accurately time heart beats and estimate heart rate variability. An overview of our method is illustrated in Figure 5.1.

State of datasets As the field of remote photo-plethysmography receives advances and the accuracies of rPPG methods improve, the demand for thorough, challenging and realistic datasets arise. Datasets originally created for alternate uses (e.g. psychological studies) often get re-purposed for rPPG analysis, which has unintended drawback (video compression, occlusion, etc.) [12, 13]. A large proportion of research in this field end up using self-recorded private datasets [14], due to which the results cannot be directly compared with prior work. These factors hinder proper development, evaluation and benchmarking of rPPG methods. To help alleviate this, we introduce a new publicly available multi-modal video dataset specifically designed to aid the study of camera-based rPPG algorithms for HR and HRV analysis.

Ground truth signals Another significant but overlooked complication with existing rPPG datasets arises from their provided ground truth signals, typically photoplethysmogram (PPG) or electrocardiogram (ECG) waveform. These signals are often plagued by artefacts, for example in the form of large spikes caused by unwanted sensor movement and interference [3, 15]. This causes false peak detections resulting in incorrect ground truth HR and HRV measures, thereby leading to unreliable evaluation. Additionally, directly utilizing

raw PPG/ECG signals as ground truth leads to another issue in the evaluation process: evaluatees are free to choose the method of obtaining peaks on them, and they could use the very same peak detection algorithm that is used by the method under evaluation on the rPPG signal. Since the estimated rPPG peaks are evaluated against ground truth peaks generated by the same algorithm, this can lead to a "detector-bias" in the computed error measures. For example, an algorithm that blindly detects a fixed number of heart-beats on all signals will incorrectly report zero error.

To help solve these problems caused by noisy ground truth signals, we introduce CleanerPPG: a public collection of human-verified peak/heart-beat annotations on ground truth signals of existing publicly available rPPG datasets. This makes the ground truth heart-beats absolute and independent, and therefore offers more accurate, fairer and standardized evaluations and benchmarking. CleanerPPG is made publicly available[1] and intended to develop into a continuously growing community-driven collection for all future datasets.

Contributions We make the following contributions in this work:

(i) We present an efficient unsupervised rPPG pipeline that can estimate heart-rate from RGB webcams. This method has the advantage that it does not require any rPPG specific training and it can perform its analysis with real-time speeds.

(ii) Our method is able to time individual heart beats in the estimated pulse signal to compute heart rate variability. This body of work has received little attention, and we set the first benchmarks on multiple public datasets.

(iii) We perform an in-depth HR and HRV evaluation on an exhaustive collection of 13 public and self-recorded datasets exploring a varied range of unique facets. We also demonstrate state-of-the-art level performance on six public datasets.

(iv) We introduce a new publicly available high frame-rate dataset, VicarPPG 2, specifically designed to evaluate rPPG algorithms under various subject conditions for HR and HRV analysis.

(v) Lastly, we tackle the problem of noisy ground truth signals and the peak detector bias by releasing a collection of hand-cleaned heart-beat peaks for existing public datasets.

5.2 Related Work

Signal processing based rPPG methods Since the early work of Verkruysse *et al.*[5], who showed that heart rate could be measured from consumer grade camera recordings in ambient light, a large body of research has been conducted on the topic. Extensive reviews of these work can be found in [4, 14, 16]. Most published rPPG methods work either by applying skin detection on a certain area in each frame or by selecting one or multiple regions of interest and track their averages over time to generate colour signals. A general division can be made into methods that use blind source separation (ICA, PCA) [17–19] vs those that use a 'fixed' extraction scheme for obtaining the blood volume pulse (BVP)

[1] This dataset can be requested via the link: `www.vicarvision.nl/datasets/cleanerppg`

signal [20–24]. The blind source separation methods require an additional selection step to extract the most informative BVP component signal. To avoid this, we make use of a 'fixed' extraction scheme in our method.

Among the 'fixed' methods, multiple stand out and serve as inspiration and foundation for this work. Tasli *et al.* [22] presented the first face modelling based signal extraction method and utilized detrending [25] based filtering to estimate BVP and heart rate. The CHROM [20] method uses a ratio of chrominance signals which are obtained from RGB channels followed by a skin-tone standardization step. Li *et al.* [21] proposed an extra illumination rectification step using the colour of the background to counter illumination variations. The SAMC [23] method proposes an approach for BVP extraction in which regions of interest are dynamically chosen using self adaptive matrix completion. The Plane-orthogonal to skin (POS) [24] method improves on CHROM. It works by projecting RGB signals on a plane orthogonal to a normalized skin tone in normalized RGB space, and combines the resulting signals into a single signal containing the photopleytismographic information. We take inspiration from Tasli *et al.* [22] and further build upon POS [24]. We introduce additional signal refinement steps for accurate peak detection to further improve HR and HRV analysis.

Deep learning based rPPG methods Most recent works have applied deep learning to extract either heart rate or the blood volume pulse directly from camera images. They rely on the ability of deep networks to *learn* which areas in the image correspond to heart rate. This way, no prior domain knowledge is needed and the system learns the underlying rPPG mechanism from scratch. DeepPhys [6] is the first such end-to-end method to extract heart and breathing rate from videos. HR-CNN [7] uses two successive convolutional neural networks (CNNs) [26] to first extract a BVP from a sequence of images and then estimate the heart rate from it. RhythmNet [27] uses a CNN and gated recurrent units to form a spatiotemporal representation for HR estimation. The recent work of AutoHR [28] employs neural architecture search to discover temporal difference convolution as a strong backbone to capture the rPPG signal from frame sequences. These methods have shown state-of-the-art performance on multiple public and private datasets. Our presented algorithm is unsupervised and makes use of an active appearance model [29] to select regions of interest to extract a heart rate signal from. Due to this, no rPPG specific model training is required while prior domain knowledge is more heavily relied upon.

Heart Rate Variability from rPPG Some past methods have also attempted extracting heart rate variability from videos [19, 30, 31]. A good overview of this is provided by Rodriguez *et al.* [32]. Because HRV is calculated based on variations in inter-beat intervals, it is crucial that single beats are detected and localized with a high degree of accuracy. Methods that otherwise show good performance in extracting HR can be unsuitable for HRV analysis since they may not provide beat locations. Rodriguez *et al.* [32] evaluate their baseline rPPG method for HRV estimation. Their method is based on bandpass filtering the green channel from regions of interest. However, their results are only reported on their own private dataset (not publicly available), which makes direct comparison difficult. More recent works have shown and benchmarked video-based HRV measurement on publicly available datasets. Finžgar and Podržaj [33] introduce a wavelet transform and custom

inter-beat-interval filtering rPPG algorithm and evaluate it on the publicly available PURE dataset [34]. They shown good correlation between time-domain ultra-short term HRV measurements from rPPG and PPG. Work by Li *et al.* [35] highlight the effectiveness of a clear signal for peak detection, and apply a slope sum function to create more pronounced peaks in the rPPG signal. Concurrent work by Song *et al.* [36] introduces one of the first deep learning based fully supervised techniques for HRV estimation, relying on a generative adversarial network to learn denoising of rPPG signals. Both these papers report their results on the UBFC-RPPG dataset [37], which is publicly available. Our method also estimates heart rate variability by obtaining precise temporal beat locations from the filtered BVP/rPPG signal, and we report our HRV results on a large number of public datasets.

rPPG datasets Due to a scarcity of rPPG datasets in the past, initial attempts at evaluating rPPG methods were on private self-recorded videos [24, 30, 38]. Some of the earliest publicly available datasets with heart rate annotations repurposed for rPPG research were introduced in [12] and [13], both of which were originally recorded for the purpose of psychological studies. Although the lab setting and video compression makes some of these datasets less than ideal for rPPG, their public availability and large sample sizes provide a common platform for benchmarking rPPG methods. More recently, [22, 34, 39, 40] introduced datasets specifically recorded with the intention of being used for remote heart rate estimation research. These sets include variations in illumination, physical/physiological conditions, and camera types. In this chapter, we introduce a new high frame-rate video dataset for rPPG evaluation designed with a focus on evaluating short-term heart rate variability estimation, which require longer observations. This dataset is made publicly available for research use[2].

5.3 Method

We present a method for extracting heart rate (HR) and heart rate variability (HRV) from the face in real-time using only a consumer grade webcam and CPU, as shown in Figure 5.1.

5.3.1 Skin pixel selection

The first step in the pipeline includes face finding [41] and fitting an active appearance model (AAM) [29]. This AAM is then used to determine facial landmarks (from the AAM shape vector) as well as the head orientation (by measuring angular deviation from the mean frontal pose). The landmarks are used to define a region of interest (RoI) which only contains pixels on the face belonging to skin. This allows us to robustly track the pixels in this RoI over the course of the whole video. Our RoI consists of the upper region of the face excluding the eyes (determined empirically). An illustration of this can be seen in Figure 5.1. The head orientation is used to measure and track the pitch, roll, and yaw angles of the head per frame. Across all pixels in the RoI, the averages for each colour channel (R,G,B) is computed and tracked (concatenated) to create three colour signals.

[2]This dataset can be requested via the link: `www.vicarvision.nl/datasets/vicarppg2`

5.3.2 Signal extraction

The colour signals and the head orientation angles are tracked over a running time window of 8.53 seconds, which corresponds to 256 frames at 30 fps, or 512 frames at 60 fps. To counteract the impact of variations in frame rates of the input, all signals are resampled (using linear interpolation) to a fixed sampling rate of 30 or 60 Hz, whichever is closer to the frame rate of the source video. The choice of this window duration and sampling rate is based on the resulting signal length being a power of two, which is compatible with optimized fast Fourier transform operations. Subsequently, the three colour signals from R, G and B channels are combined into a single rPPG signal using the POS method [24]. This method filters out intensity variations by projecting the R, G and B signals on a plane orthogonal to an empirically determined normalized skin tone vector. The resulting 2-D signal is combined into a 1-D signal via a weighted sum with the weight determined by the ratio of standard deviations of the two signals. This ensures that the resulting rPPG signal contains the maximum amount of the pulsating component.

5.3.3 Signal filtering

Rhythmic motion noise suppression

A copy of the extracted rPPG signal as well as the head-orientation signals are converted to the frequency domain using Fast Fourier Transform. The three resulting head-orientation spectra (one each of pitch, roll, and yaw) are combined into one via averaging. This is then subtracted from the raw rPPG spectrum after amplitude normalization. This way, the frequency components having a high value in the head-orientation spectrum are attenuated in the rPPG spectrum. Subsequently, the frequencies outside of the human heart rate range (0.7 - 4 Hz / 42 - 240 bpm) are removed from the spectra.

Wide & narrow band filtering

The strongest frequency component inside the resulting spectrum is then used to determine the passband range of a narrow-bandpass filter with a bandwidth of 0.47 Hz. This bandwidth has been chosen empirically and depends on the robustness of the subsequent peak detection algorithm to distinguish heat beat peaks from noise (higher the robustness, wider this bandwidth can be). This bandpass filter can either be realized via inverse FFT or a high order FIR filter (e.g. ~50^{th} order Butterworth). The selected filter is then applied to the original extracted rPPG signal to produce noise-free BVP.

5.3.4 Post processing

To prevent minor shifts in the locations of the crest of each beat over multiple overlapping running windows, the signals from each window are overlap added with earlier signals [20, 24, 42]. First, the filtered rPPG signal is normalized by subtracting its mean and dividing it by its standard deviation. During resampling of the signal, the number of samples to shift is determined based on the source and resampled frame rates. The signal is then shifted back in time accordingly and added to the previous/already overlapped signals. Older values are divided by the times they have been overlap added, to ensure all temporal locations lie in the same amplitude range. Over time, a cleaner rPPG signal is obtained from this.

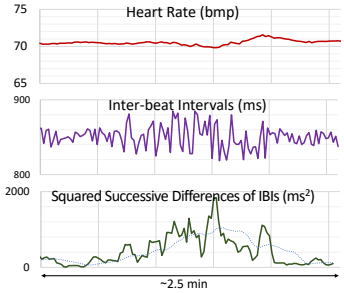


Figure 5.2: Example of heart rate variability computation: Even when the heart rate (HR) is almost constant, the underlying inter-beat intervals (IBIs) can have many fluctuations. This is detected by rising squared successive differences (SSD), a measure of heart rate variability.

5.3.5 Output calculation

Once a clean rPPG signal is obtained, we can perform peak detection on it to locate the individual beats in time in the signal. From the located beats, heart rate and heart rate variability can be calculated. To do this, we first extract the inter-beat-intervals (IBIs) from the signal, which are the time intervals between consecutive beats.

Inter-beat interval pre-filtering

Before calculating HR/HRV, the extracted inter-beat intervals (IBI) are filtered to remove noise caused by false positive/negative peak detections. First, all IBIs lying outside the range of 250 ms to 2000 ms are excluded (corresponding to the human heart rate range of 30 to 240 bpm). To further remove strong outliers from the signals, intervals farther than three standard deviations from the mean are removed.

Heart rate calculation

Heart rate is calculated by averaging all IBIs over a time window, and computing the inverse of it. That is, $\mathrm{HR}_w = 1/\overline{\mathrm{IBI}}_w$, where $\overline{\mathrm{IBI}}_w$ is the mean of all inter-beat intervals that fall within the time window w. This gives the heart rate in Hertz (assuming IBIs in seconds), and multiplying by 60 gives us the heart rate in beats-per-minute. The choice of this time window can be based on the user's requirement (e.g. instantaneous HR, long-term HR).

Heart rate variability calculation

Multiple metrics can be computed to express the measure of heart rate variability in different units. In this work, we focus on one of the most commonly used time-domain metric for summarizing HRV called the 'root mean square of successive differences' (RMSSD) [11, 32, 38, 43], expressed in units of time. As the name suggests, this is computed by calculating the root mean square of time difference between adjacent IBIs:

$$\mathrm{RMSSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathrm{IBI}_i - \mathrm{IBI}_{i+1})^2}, \tag{5.1}$$

where IBI_i represents the i^{th} inter-beat interval, and N represents the number of IBIs in the sequence. A graphical example of such HRV calculation is shown in Figure 5.2. Because RMSSD is more susceptible to noise, only IBIs within the first standard deviation around the mean are considered. Along with RMSSD, we calculate another time-domain HRV

metric known as the 'standard deviation of NN intervals' (SDNN) [11], which is simply the standard deviation of all filtered IBIs in the sequence.

In addition, we also compute two frequency-domain metrics of HRV, simply known as 'low-frequency' (LF) and 'high-frequency' (HF) bands [43] (as well as a ratio of them), that are commonly used in rPPG HRV literature [17, 30, 32]. The LF and HF components are calculated using Welch's power spectral density estimation [44]. Since Welch's method expects evenly sampled data, the IBIs are interpolated at a frequency of 4Hz using spline interpolation and detrended to remove very low frequency components [25]. The power of each band is calculated as total power in a region of the periodogram: the LF band from [0.04 to 0.15 Hz], and the HF band from [0.15 to 0.4 Hz]. Both metrics are converted to normalized units by dividing them by the sum of LF and HF. Details about these metrics can be found in [11, 45].

5.4 Datasets

To compare against prior work and study the properties of the proposed method, we evaluate on a comprehensive collection of datasets. Table 5.3 provides summarized details of existing datasets used in this chapter, while the self-recorded ones are listed below. Some example frames from these self-recorded datasets are shown in Figure 5.4.

StableSet rPPG dataset To make a proof-of-concept test of our proposed rPPG method, the StableSet rPPG dataset was collected. This dataset contains recordings of participants while they were shown short video clip stimuli on a screen facing them. During the recording of this dataset, the participating subjects' head movements were physically stabilized using a chin rest with the intention of minimizing motion induced noise in rPPG measurements. A total of 24 participants were included in this dataset, aged between 18 and 30 years (with a mean of 21.5 years), and having a male : female gender ratio of 9 : 15. Ground Truth was collected in the form of ECG (via a Mobi8 device) at a sampling rate of 1 KHz, and the videos were recorded using a front-facing HD camcorder (JVC GZ-VX815) with a resolution of 1920×1080 pixels at a framerate of 25 fps. Camera settings like brightness, aperture, backlight compensation and white balance were set to manual on

PURE [34] 10 subjects / 59 videos Subjects recorded with varying movement patterns: talking, slow/fast translation, and small/large rotation. 480p @ 30fps PNG images [lossless] Ground truth PPG @ 60Hz	**UBFC-RPPG** [37] 42 subjects / 42 videos Subjects recorded while playing a stressful game (under "realistic" partition). 480p @~30fps Raw video format [lossless] Ground truth PPG @ 30/60Hz	**MMSE-HR** [46] 40 subjects / 102 videos Part of a larger multi-modal corpus containing recordings while subjects exhibit facial expressions. 1040x1392p @25fps JPEG images Instantaneous HR @ 1KHz	**VIPL-HR** [39] 107 subjects / 2378 videos Large dataset with a range of movement, illumination, and camera types. ~460x502p face crops 25~30fps – MJPEG format Ground truth PPG @ 60Hz	**ECG-Fitness** [7] 17 subjects / 17 videos Recordings while exercising on rower, bike and elliptical equipment, and also while talking. 1080p @ 30fps YUV format [lossless] Ground truth ECG @ ~125Hz
MAHNOB-HCI [12] 27 subjects / 527 videos Subjects recorded while watching video stimuli. 780×580p @ 61fps H.264 format Ground truth ECG @ 256Hz.	**VicarPPG** [22] 10 subjects / 20 videos Unrestrained subjects recorded before and after performing strenuous workout. 720p @ ~30fps [variable] H.264 format Ground truth PPG @ 30Hz.	**DEAP** [13] 874 videos of 22 subjects. Subjects recorded while watching music videos. Faces are significantly occluded by electrodes. 720x576p @ 50fps H.264 format Ground truth PPG @ 128Hz.	**MoLi-PPG** [40] 170 videos of 30 subjects. Subjects recorded under varying illumination, movement, and speech. 1080p/720p/600p @ 25/50fps MPEG-4 Part 2 format Ground truth ECG @ 256Hz.	**COHFACE** [47] 164 videos of 40 subjects. Subjects recorded illuminated by a spotlight and by uneven natural light. 480p @ 20fps MPEG-4 Part 2 format Ground truth PPG @ 256Hz.

Figure 5.3: A list of the previously available rPPG datasets used in the this chapter, along with their key details.

Figure 5.4: Example images from newly introduced datasets: (left to right) VicarPPG 2, EatingSet, and StableSet. The example from VicarPPG 2 shows subject suddenly turning their head. The EatingSet image shows subject taking a long sip resulting in face occlusion. The subjects in StableSet were physically stabilized using the shown chin rest (face removed for privacy reasons).

the camcorder to keep filming conditions ideal and constant. Prior informed consent was obtained from all participants. However, due to privacy restrictions within the consent, the videos of this dataset are not made available publicly.

VicarPPG-2 dataset Specifically aimed at evaluating rPPG algorithms at estimating heart rate and short-term heart rate variability (which requires a minimum observation of 5 minutes [11]), we recorded the VicarPPG-2 Dataset. 10 subjects participated in the data collection. The male : female ratio was $7:3$ with an average age of 29 ± 5 years, and skin types ranging on a Fitzpatrick scale [48] from II to IV.

Participants were asked to sit in front of a computer screen (~1 meter distance) on which the instructions were shown, a webcam was mounted on top of the screen and an LED ring lamp was mounted behind the camera. Screen brightness was reduced as far as possible to minimize the influence of screen light on the face. All videos were recorded using a Logitech Brio webcam at a fixed framerate of 60 fps using an H.264 compliant encoder (Microsoft Media Foundation), and stored in `mp4` containers. The recording location was illuminated by natural ambient light in addition to a LED ring lamp (Falcon Eyes DVR-300DVC) to prevent strong shadows and influences of large changes in natural light. The ground truth signals were recoded in the form of synchronized ECG signals at 250 Hz sampling frequency obtained via an Arduino based ECG board (AD8232), and synchronized PPG signals at 60 Hz obtained via a pulse oximeter device (CMS50E) attached to the left index finger of the participant. The following four scenarios/conditions were recorded for each participant:

(i) Baseline: Participants sitting naturally while watching a relaxing video or reading an article on screen.

(ii) Movement: participants performing four different types of pre-planned angular body/head movements: turning head side-to-side (shaking), moving head up and down (nodding), a combination of head shaking and nodding (round), moving eyes while keeping head still, and naturally bobbing their heads while listening to music (dance).

(iii) Stress: Participants playing a stress-inducing Stroop effect [49] based game.

(iv) Post-workout: Participants sitting unrestrained after performing fatigue-inducing physical workouts to induce higher heart rates.

Each condition was recorded for a duration of 5 minutes to allow for the computation of short-term heart rate variability, a total of 200 minutes of video were collected. Out of 400 collected ground truth files, two had to be removed from the dataset due to excessive finger movement in the PPG device and gradual detachment of the ECG ground electrode, leading to unusable signals. This dataset stands out as it was explicitly collected for RPPG purposes featuring 5-minute long 60 fps camera recordings under various physical/physiological conditions, with simultaneous ECG and PPG ground truth recordings. Informed consent was obtained from all participating subjects. This dataset is available for research purpose, and can be requested via the link: `www.vicarvision.nl/datasets/vicarppg2`.

EatingSet rPPG dataset This is a self-recorded video dataset comprising of 20 subjects, with a male : female ratio of 14 : 6. The average age in the dataset was 32 ± 8.6 years, and the subjects had skin types ranging on the Fitzpatrick scale [48] from II to IV. The recording setup and conditions were similar to those of VicarPPG 2. Participants were sitting at 1 meter distance to a screen on which instructions were shown, while being illuminated by an LED ring lamp. All videos were recorded using a Logitech C920 webcam at 30 fps in uncompressed YUYV422 pixel format. PPG signals were collected as ground truth, at a frequency of 60 Hz, via a pulse oximeter device (CMS50E) attached to the left index finger of the participant. During recording participants were asked to consume 4 types of food items with varying consistency. These include a sip of water (drink), a cookie (crumbly), a marshmallow (chewy) and multiple almonds (hard). Informed consent was obtained from all participants. However, due to privacy restrictions, the videos of this dataset are not made available publicly. This unique dataset contains a variety of natural, non-rhythmic deformations and facial occlusions while eating as they might occur in real-world situations, and serves as a challenging testbed for rPPG evaluation.

CleanerPPG ground truth dataset Finally, we introduce a meta-dataset which contains cleaned ground truth signals of already existing datasets. All peaks/beats detected from ground truth signals (ECG, PPG) of all datasets used in this chapter were hand-verified and corrected by human expert annotators, in order to achieve a more accurate and standardized evaluation.

To do this, candidate peaks were first obtained using a gradient-based signal peak detector [50]. Following this, a human verification step was performed wherein an expert matched, verified and corrected every candidate peak by observing the shape of the raw PPG/ECG waveform and the inter-beat intervals. A specialized Python tool was developed for this step, which allow annotation of single peaks as well as zooming and an overview of the resulting RR intervals. Peaks were annotated at the crest of the PPG signal or at the highest point of the R peak in the ECG signal. Ectopic beats (genuine extra heart beats that can occur between two regular beats) in the ECG signal were included for the sake of completion. Parts of the signal that were too noisy due to movement of electrodes or finger to make out clear peaks were left blank.

These steps led to the removal of false positive and negative peak detections caused by artefacts in the signal, resulting in a collection of noise-free ground truth heart beat annotations for an exhaustive set of publicly available rPPG datasets. Annotators spent an average of ~30 seconds per minute of signal duration, results in a total of 36 person-hours of annotation work for cleaning 75 hours of ground truth data. These annotations can be used to perform a more accurate evaluation of rPPG methods, especially for noise-sensitive measures like HRV. These peak annotations can also be used for training fully supervised machine learning methods to be able to distinguish between true heat beat peaks and noise, thereby improving accuracy of such methods. This collection is available for research purposes, and can be requested via the link: `www.vicarvision.nl/datasets/cleanerppg`. All experiments performed in this chapter utilize these hand-cleaned peaks.

5.5 Experiments and Results

5.5.1 Impact of ground truth peak cleaning: CleanerPPG

To lay foundation for the rest of the experiments in this chapter, we first study the impact and value of evaluating rPPG methods against hand-cleaned ground truth. Figure 5.5 provides examples of the kind of artefacts that plague raw ground truth PPG/ECG signals resulting in incorrect peak detections. To provide a more qualitative analysis, we measure the agreement between the raw ground truth peaks (obtained via a gradient-based peak detector [50]) and the hand-cleaned ground truth peaks from CleanerPPG. We do this by computing the mean absolute error/difference between their computed HR and HRV values on all datasets. The results of this study can be seen in Table 5.1.

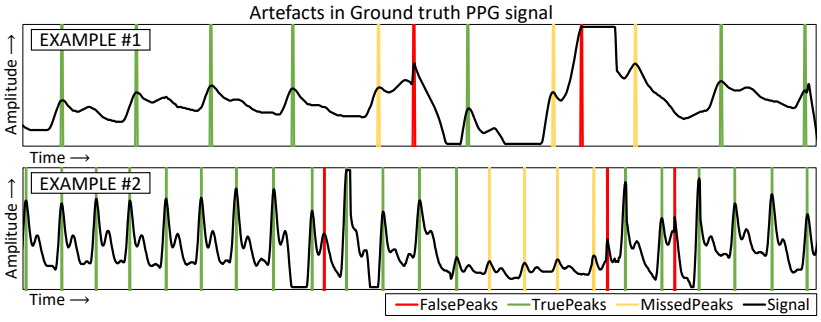


Figure 5.5: Examples of peak detection on artefact-prone raw ground truth PPG signals. Spikes in the amplitude result in false peak detections while some regions of the signal are attenuated resulting in true peaks not being detected.

Dataset	HR (bpm)	HRV [RMSSD] (ms)
PURE	0.34 ±1.8	7.9 ±13.1
UBFC-RPPG	0.24 ±0.6	11.2 ±8.4
MMSE-HR	3.57 ±3.5	—
VIPL-HR	0.57 ±3	17.33 ±57.9
COHFACE	0.46 ±2.2	14.36 ±49.6
ECG-Fitness	3.47 ±15.2	19.24 ±76.1

Dataset	HR (bpm)	HRV [RMSSD] (ms)
MoLi-PPG	1.6 ±9.3	4.4 ±16.1
DEAP	0.41 ±1.7	11.85 ±37.4
VicarPPG	0.01 ±0.02	0.11 ±0.3
VicarPPG 2	0.08 ±0.5	4.52 ±19.4
EatingSet	4.33 ±1.6	6.54 ±10.1
StableSet	1.11 ±6.1	10.15 ±43.6
MAHNOB	0.27 ±1.8	3.31 ±23.0

Table 5.1: Deviations (in terms of mean absolute error) in HR (bpm) and HRV [RMSSD] (ms) calculations between the raw ground truth and the hand-cleaned peaks from the CleanerPPG set. Many datasets exhibit notable deviations due to artefacts in the raw signal. This leads to miscalculation of error metrics used for evaluation.

On average over all datasets, the heart rate values computed from the raw ground truth peaks deviate from the clean peaks in the range of 0.01 to 4.3 bpm, with almost all datasets containing videos that deviate over 10 bpm. For some datasets (VicarPPG {1&2}, UBFC, MAHNOB, PURE, COHFACE, VIPL, DEAP), the raw PPG/ECG signals provided are already of fairly good quality with few artefacts, resulting in a relatively small average deviation from the cleaned peaks (less than ~0.5 bpm). On the other hand, many datasets exhibit significant deviations from the cleaned ground truth (MoLi-PPG, StableSet), with some diverging quite heavily (EatingSet, MMSE-HR, ECG-Fitness). Similar but larger deviations are also observed for HRV (RMSSD) calculations, which is much more sensitive to fine errors in peak detection.

These deviations can be problematic when comparing/benchmarking the performance of rPPG methods against each other. This is especially the case when the gap in performance between methods is small. For example, the raw ground truth of PURE deviates from cleaned ones by 0.36 bpm. While small, this is significant since the top two state-of-the-art methods have a performance gap of only 0.3 bpm on this dataset (as seen in Table 5.2).

These results suggest that using the raw ground truth peaks for evaluation can result in substantial miscalculations of error metrics, potentially leading to incorrect conclusions. In all the experiments that follow, the cleaned CleanerPPG peaks are used as the ground truth.

5.5.2 Benchmarking and comparison against state-of-the-art

In order to study the generalizability of the proposed rPPG method, we benchmark it's performance on a range of datasets. To assess accuracy, we measure the deviation of the predicted HR/HRV measures from the ground truth in terms of mean absolute error (MAE), which is the average of the absolute differences between predicted and true values (obtained within a set time window for HR). The results for heart rate and heart rate variability analysis are listed in Tables 5.2 and 5.3 respectively. These results are also compared against prior work to provide context w.r.t. the state of the art. In addition, results of a blind baseline estimator that always predicts a heart rate of 75 bpm (mean HR of all datasets) are also included in the table for additional context.

Heart Rate Estimation

The results for heart rate analysis are listed in Tables 5.2. Here, while comparing, it should be noted that the supervised methods train or fine-tune on part of the dataset they

	Method	PURE	UBFC-RPPG	MMSE-HR	VIPL-HR	COHFACE	ECG-FITNESS	MAHNOB-HCI	MoLi-PPG	DEAP	VicarPPG	VicarPPG 2	EatingSet	StableSet
	Baseline (75 bpm)	17.6 ±15.1	13.7 ±9.1	12.6 ±12.4	9.16 ±8.5	10.7 ±5.8	35.3 ±21.6	9.71 ±6.6	8.98 ±8.3	8.03 ±5.7	16.2 ±10.8	8.38 ±6.1	10.1 ±7.2	7.9 ±6.1
Unsupervised Signal Processing Methods	FaceRPPG [Ours]	0.36 ±0.4 {15} 0.26 ±0.3 {30} 0.22 ±0.2 {∞}	2.6 ±6.7 {15} 2.37 ±7.1 {30} 1.98 ±6.2 {∞}	3.82 ±10.1 {15} 3.73 ±10.1 {30} 3.89 ±10.3 {∞}	10.6 ±11.5 {15} 10 ±11 {30} 9.88 ±10.9 {∞}	12 ±9.3 {15} 10.8 ±9.8 {30} 10.4 ±9.5 {∞}	24.9 ±22.9 {15} 24.9 ±22.4 {30} 24.8 ±22 {∞}	16.9 ±13.5 {15} 13.1 ±10.4 {30} 13.1 ±10.4 {∞}	4.85 ±7.2 {15} 4.45 ±7.1 {30} 3.5 ±6.9 {∞}	7.17 ±5.9 {15} 6.24 ±6.1 {30} 5.57 ±5.5 {∞}	1.83 ±3.5 {15} 1.28 ±2.2 {30} 1.53 ±3.5 {∞}	3.43 ±5.1 {15} 3.09 ±5.0 {30} 2.59 ±4.3 {∞}	3.18 ±2.4 {15} 2.97 ±2.5 {30} 2.77 ±2.7 {∞}	0.9 ±1.6 {15} 0.7 ±1.6 {30} 0.59 ±1.6 {∞}
	Green/EVM [8]	—	—	—	—	—	—	—	—	—	5.6 {15} [22]	—	—	—
	Tasli *et al.* [22]	—	—	—	—	—	—	—	—	—	4.2 {15}	—	—	—
	C-MCCA [51]	—	—	—	—	—	—	—	—	3.66	—	—	—	—
	cPR+fine [40]	—	2.1	—	---	—	—	—	6.13	—	—	—	—	—
	POS [24]	—	4.73 {30} [52]	5.77 {15} [52]	11.5 {10} [27]	—	—	—	—	—	—	—	—	—
	ICA [53]	*24.1 [54]	6.02 {30} [52]	5.84 {15} [52]	—	—	—	—	—	—	—	—	—	—
	NMD-HR [54]	*8.68	—	—	—	—	—	—	—	—	—	—	—	—
	2SR [55]	*2.44 {∞}	—	—	—	*20.98 {∞} [7]	*43.66 {∞} [7]	*13.84 {∞} [7]	—	—	—	—	—	—
	CHROM [20]	*2.07 {∞}	3.7 {30} [52]	5.59 {15} [52]	≤16.9 {10} [27]	*7.8 {∞} [7]	*21.37 {∞} [7]	*13.49 {∞} [7]	—	—	—	—	—	—
	LiCVPR [21]	*28.2 {∞}	—	≤19.95 {10} [27]	—	*19.98 {∞} [7]	*63.25 {∞} [7]	*7.41 {∞} [7]	—	—	—	—	—	—
	Wavelet [56]	—	—	2.4 {10}	—	—	—	—	—	—	—	—	—	—
	PVM [52]	—	4.47 {30}	4.38 {15}	—	—	—	—	—	—	—	—	—	—
	MAICA [57]	—	3.43	3.91	—	—	—	—	—	—	—	—	—	—
	VD+LMS+HRE [58]	—	—	—	25.52	—	—	—	—	—	—	—	—	—
	CK [59]	—	2.29	—	—	—	—	—	—	—	—	—	—	—
Supervised ML Methods	HR-CNN [7]	*1.84 {∞}	—	—	—	*8.1 {∞}	*14.48 {∞}	*7.25 {∞}	—	—	—	—	—	—
	SAMC [23]	—	—	≤*11.37	15.9 {10} [27]	≤6.23	—	—	—	—	—	—	—	—
	ST-CNN [60]	*0.87	—	—	—	—	—	—	—	—	—	—	—	—
	Attentn-CNN [61]	—	—	—	—	—	*6.8	—	—	—	—	—	—	—
	DeepPhys [6]	—	—	—	*11 {10} [27]	—	—	4.57 {∞}	—	—	—	—	—	—
	RythmNet [27]	—	—	≤*5.03 {10}	*5.3 {10}	—	—	≤*4.00 {∞}	—	—	—	—	—	—
	Siamese-rPPG [62]	*0.63	*0.48	—	—	—	—	—	—	—	—	—	—	—
	AutoHR [28]	—	—	5.87	*5.68	—	—	*3.78 {∞}	—	—	—	—	—	—
	PhysNet [63]	—	—	≤13.25 [28]	*10.8 [28]	—	—	5.96 {∞}	—	—	—	—	—	—
	PulseGAN [36]	—	2.09	—	—	—	—	—	—	—	—	—	—	—

Table 5.2: A comparison of the performances of various methods in terms the mean absolute error in beats per minute (bpm). {15}, {30}, and {∞} represent the HR calculation windows of 15 s, 30 s, and full-video length respectively. * represent accuracies obtained on a smaller (test) subset of the full dataset. ≤ represents root mean squared error, which is always greater than or equal to mean absolute error. Baseline represents the accuracy obtained by always predicting a heart rate of 75 bpm (average HR over all datasets). The reported results of the proposed FaceRPPG method (and the baseline) are against the cleaned ground truth from the CleanerPPG set. The references next to the results denote the source from which they were obtained. The different colours separate the methods into two different categories: unsupervised signal processing methods, and fully supervised deep learning methods. Note that the supervised methods also train or fine-tune their parameters on parts of dataset they are being evaluated on, while unsupervised methods do not. The proposed method outperforms most prior work, including fully supervised ones.

Method	HRV Metric	PURE	UBFC	VIPL-HR	COHFACE	ECG-FITNES	MAHNOB	MoLi-PPG	DEAP	VicarPPG	VicarPPG 2	EatingSet	StableSet
FaceRPPG[Ours]	RMSSD (ms)	15 ±12.7	16 ±22.5	73 ±57.8	119 ±44.4	82 ±53.2	108 ±51.4	43 ±43.7	74 ±41.4	22 ±13.8	26 ±18.9	37 ±33	21 ±37.9
	SDNN (ms)	18 ±10.4	19 ±14.9	49 ±45.5	80 ±39.6	53 ±48.2	107 ±51.8	36 ±30.2	46 ±35.2	44 ±28.1	16 ±15.3	16 ±11.5	22 ±23.8
	LF & HF(n.u.)	0.1 ±0.1	0.2 ±0.13	0.3 ±0.18	0.2 ±0.17	0.3 ±0.2	0.3 ±0.2	0.2 ±0.13	0.3 ±0.18	0.1 ±0.1	0.2 ±0.11	0.3 ±0.18	0.1 ±0.1
	LF/HF	1.3 ±3.03	1.0 ±0.99	1.6 ±2.8	2.4 ±8.01	3.2 ±4.37	2.9 ±11.8	1 ±1.22	1.9 ±2.76	0.5 ±0.36	1.5 ±1.43	1.5 ±1.44	0.5 ±0.76
Finžgar *et. al.*[33]	RMSSD (ms)	16.77	—	—	—	—	—	—	—	—	—	—	—
	SDNN (ms)	8.14	—	—	—	—	—	—	—	—	—	—	—
SSF[35]	RMSSD (ms)	—	47	—	—	—	—	—	—	—	—	—	—
	SDNN (ms)	—	25	—	—	—	—	—	—	—	—	—	—
CHROM[20]	RMSSD (ms)	—	93 [35]	—	—	—	—	—	—	—	—	—	—
	SDNN (ms)	—	38.9[36]	—	—	—	—	—	—	—	—	—	—
PulseGAN[36]	SDNN (ms)	—	24.3	—	—	—	—	—	—	—	—	—	—

Table 5.3: Heart rate variability estimation performance in terms of mean absolute error. The metrics included are RMSSD and SDNN (in milliseconds), LF and HF (in normalized units), and the ratio of LF/HF. The different colours separate the different metric types as denoted in the second column. The proposed FaceRPPG method outperforms all prior work, including fully supervised ones. It performs well on datasets with low video compression noise or limited subject movement, but fails when these factors become large. FaceRPPG results are reported against the cleaned ground truth from the CleanerPPG set.

are evaluated on, while unsupervised methods do not. Further note that the results for most supervised methods and some unsupervised methods (marked with a *) are reported on a smaller unspecified subset (test set) of the dataset. Also, the reported heart rate mean absolute errors are computed using different time-window sizes (denoted within the brackets $^{\{\}}$ when known). To aid comparison, we report our results on three most commonly used window sizes: 15 secs, 30 secs and full-video length (denoted by $^{\{\infty\}}$).

The proposed method performs well at the state-of-the-art level or beyond on most of the datasets: PURE, VIPL-HR, MoLiPPG, VicarPPG, UBFC-RPPG, MMSE-HR. The performance on these datasets is not only better than other unsupervised signal-processing methods, but also better or on par with fully supervised deep learning methods. The very high accuracy of 0.22 - 0.36 bpm on the PURE dataset can be attributed to the videos being stored in a lossless format and thus having no compression noise. The low error rate on StableSet can be attributed to the fact that subjects' movements were physically stabilized via a chin-rest (see Figure 5.4). On MMSE-HR, our method is able to perform well despite the subjects showing a range of facial expressions (e.g. laughter). This can be attributed to the robustness of the face modelling step. Conversely on DEAP, the accuracy of the algorithm is moderate, with the likely source of errors being poor face modelling due to the presence of electrodes occluding the face. On the ECG-Fitness dataset, although the proposed methods performs better or similar to other unsupervised methods, the accuracy is quite poor. This is largely caused by the extremely high intensity movements of the subjects while performing physical exercises.

On VIPL, the source of a majority of the errors are the videos recorded on a mobile device. This could be because of relatively inferior camera sensor and/or stronger compression on such a device. On the videos of MAHNOB-HCI, which are also highly-compressed, we see that our method does not achieve a very good accuracy, similar to the majority of unsupervised signal processing methods. An interesting observation is that the error produced by almost all unsupervised methods is higher than that of a dummy baseline method that blindly predicts a heart rate of 75 bpm for any input (this is also the case for VIPL and COHFACE). Only the supervised methods are able to perform better (direct comparison not always possible since fine-tuning is performed and the reported results are on an unspecified test subset). This suggests that the high compression noise distorts the pulse information in the spatial averages of skin pixels. Deep learning based methods seem to be able to somewhat overcome this, perhaps by learning to detect and filter out the spatial 'pattern' of such compression noise.

Heart Rate Variability Estimation

The task of assessing HRV is much more noise-sensitive than estimating heart rate. In Table 5.3, the results of heart rate variability estimation are listed for all datasets. Since HRV is a relatively long-term measure, these HRV metrics are computed over complete video lengths. Our unsupervised method sets the first HRV evaluation benchmarks on most of the datasets, and outperforms all previous methods on PURE and UBFC-RPPG datasets, including deep learning based fully supervised ones (*PulseGAN* [36]).

Based on HRV literature [11] and considering that the average human heart rate variability is in the range of 19-75 ms RMSSD, error rates close to or less than ~30 ms RMSSD can be considered acceptably accurate for distinguishing between broad HRV level

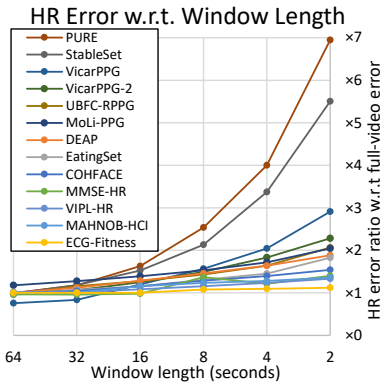


Figure 5.6: Heart rate estimation error rate (MAE) ratio with varying time-window lengths with respect to average long-term heart rate estimation error over the full video. The graph shows that the task of HR estimation becomes exponentially harder with decreasing window lengths. For example, error rate on PURE over 2-second window is 7 times worse than over the full video.

groups. Our method shows good performance on datasets that are known to have low video compression and relatively less movement (PURE, UBFC-RPPG, VicarPPG, StableSet). Reasonable performance is also obtained on some datasets containing movement (MoLi-PPG and EatingSet); while good results are obtained on VicarPPG 2 in spite of subject movement. However, accuracy is very poor on the remaining datasets that either contain high compression noise (MAHNOB-HCI, COHFACE), or exhibit large/fast movements (VIPL-HR, ECG-Fitness).

5.5.3 In-depth Analysis

Effect of window length on heart rate computation

Considering the choice of window length over which heart rate is computed is important. In principle, estimating the average heart rate over larger time windows is an easier task than estimating instantaneous heart rates over shorter windows. This is because variations in the inter-beat intervals caused by falsely detected and missed peaks average out over a large window length, thereby giving the evaluation a higher tolerance to incorrect peak detections. To illustrate this, Figure 5.6 plots the factor by which the error rate changes when using different window lengths with respect to the long-term average heart rate error over the whole video. It can clearly be seen that error rates increase exponentially with decreasing window lengths for most datasets. The datasets less affected by this are the ones on which errors caused by other factors (like movement, compression, etc.) overshadow the effect of window length (e.g. ECG-Fitness, MAHNOB-HCI). This experiment illustrates the severity by which the chosen window size can affect the heart rate estimation accuracy and this should be taken into consideration during comparative evaluations.

The rest of the experiments in this chapter are performed with a window length of 16 seconds for heart rate estimation.

Effect of lighting conditions

An important factor that leads to attenuation of the underlying blood volume pulse in the extracted rPPG signal is the illumination/lighting conditions in the video. We test our method under various lighting conditions on datasets containing labelled illumination settings: COHFACE, VIPL-HR, ECG-Fitness, and MoLi-PPG; and the results are presented

in Figure 5.7. We see that bright frontal lighting counter-intuitively leads to a degradation in performance in VIPL-HR. This is likely caused by pixel saturation, where the full colour depth of the camera cannot be exploited. Constantly flickering illumination from a computer monitor in MoLi-PPG also degrades the performance severely since such fluctuations can interfere with the pulsating component in the skin pixels. The colour temperature of the light also seems to have an influence as seen in ECG-Fitness, potentially due to dissimilar light wavelength absorption characteristics of the skin. In COHFACE, as can be expected, better performance is obtained when the room is evenly lit in comparison with uneven natural lighting. Finally, the method has good performance in the case of dark/dim lighting in MoLi-PPG and VIPL-HR.

Influence of subject movement

The physical movements of the subjects themselves can also introduce noise in rPPG extraction. This is essentially caused by the interaction of light on the observed skin pixels as it moves. The performance of our methods while subjects perform different kinds of movements are shown in Figure 5.8 on datasets containing labelled movement conditions: VIPL-HR, MoLi-PPG, ECG-Fitness, PURE, and VicarPPG 2. As per expectations, we observe that the rPPG method is most accurate when no movement is happening or when just the eyes are moving. All other kinds of movements degrade the performance somewhat, although the accuracy stay acceptable for some of them. The worst performance is observed when subjects perform large/sudden movements in multiple axis, as well as 'freestyle' head bobbing/dancing motion. In both MoLi-PPG and VicarPPG 2, an interesting observation is that vertical angular head movement (nodding) results in poorer accuracy while horizontal motion (head shaking) is handled quite well by the method. This could be due to the face modelling step being able to model side faces better than top/bottom looking faces.

Impact of rhythmic motion noise suppression The impact of explicitly detecting and reducing the head movement noise in the signal via the rhythmic motion noise suppression component (Section 5.3.3) can be gauged with the help of an ablation study. The results of such a study on VicarPPG 2 and MoLi-PPG are shown in Figure 5.9. It can be seen that the addition of the motion noise suppression component in the rPPG pipeline reduces heart rate estimation errors significantly under most movement conditions. While the reduction in error is negligible when subject movement is low (conditions steady, stable, eye movement; ~0.05 bpm), this gap in performance becomes large when subjects perform large intense head movements (~1 to ~3 bpm). The exception to this is the relatively small error difference for the left/right (MoLi-PPG) and tilting (VicarPPG 2) conditions, which can largely be attributed to poor AAM face fitting during movement resulting in incorrect head orientation measurement.

A shortcoming of the rhythmic motion noise suppression method manifests itself when the primary frequency of head movement coincides with the heart rate of the subject. In such cases, while the suppression effect from low intensity head movements do not have a dominating effect against the heart rate frequency component, high intensity head movements can. They can significantly attenuate the heart rate component in the signal causing incorrect selection of the passband range of the narrow-bandpass filter, which

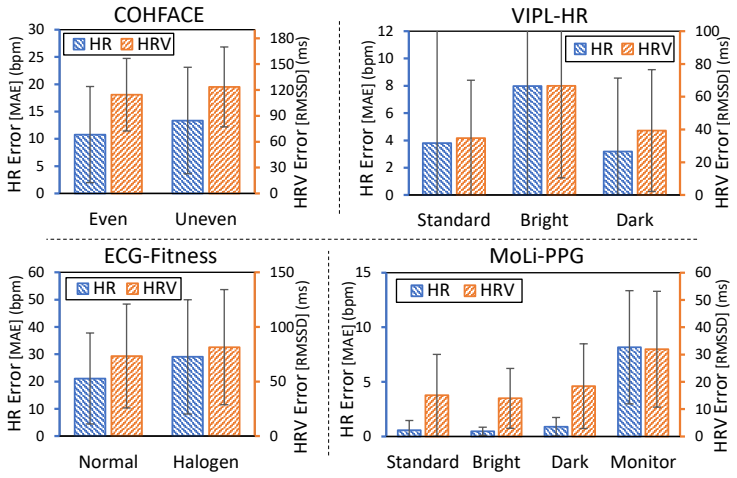


Figure 5.7: rPPG performance (MAE; error bars represent standard deviation) under different lighting conditions. Uneven illumination, oversaturation from by bright frontal lighting, yellowish glow of halogen lights, and the fluctuating monitor reflection, all somewhat degrade the performance of the rPPG method in comparison with standard ceiling lighting. However, the method performs well under dim/dark lighting.

5

can increase estimation errors. This phenomenon is observed in one of the movement conditions videos from VicarPPG 2 (subject #5).

Influence of facial activity

Another type of subject movement is the movement of facial muscles, commonly observed during talking, eating, or expressing emotions. Such movement leads to deformations in the shape of the face and stretching and moving of the skin, potentially leading to interference with the underlying BVP signal. We study the impact of facial movement on our rPPG method and present the results in Figure 5.10. The datasets included in this study are those containing labelled talking/eating condition (PURE, MoLi-PPG, VIPL-HR, EatingSet) as well as those exhibiting high facial expression activity (MMSE-HR). In all datasets, we see that talking leads to poorer performance as compared to when the face is static. Videos with higher facial arousal (an indication of facial expression activity) in MMSE-HR[3] also result in a higher error rate for HR estimation. However, while eating, chewing motion does not seem to significantly influence accuracy. In fact, closer observation revealed that occlusion of the face caused when subjects take a bike/sip seem to be the larger cause of errors. The largest duration of facial occlusion happens when subjects take a sip (glass and hand covers the face), leading to a lower HR/HRV estimation accuracy.

rPPG in high HR ranges

To study how well the accuracy of the rPPG method spans over the range of heart rates, we can compare it's performance for subjects in a rested state versus when they are in a post-workout state. This can be seen in Figure 5.11b. Furthermore, we can explicitly

[3]Facial expressions and arousal on MMSE-HR were obtained via an automated facial expression analysis tool called FaceReader [64].

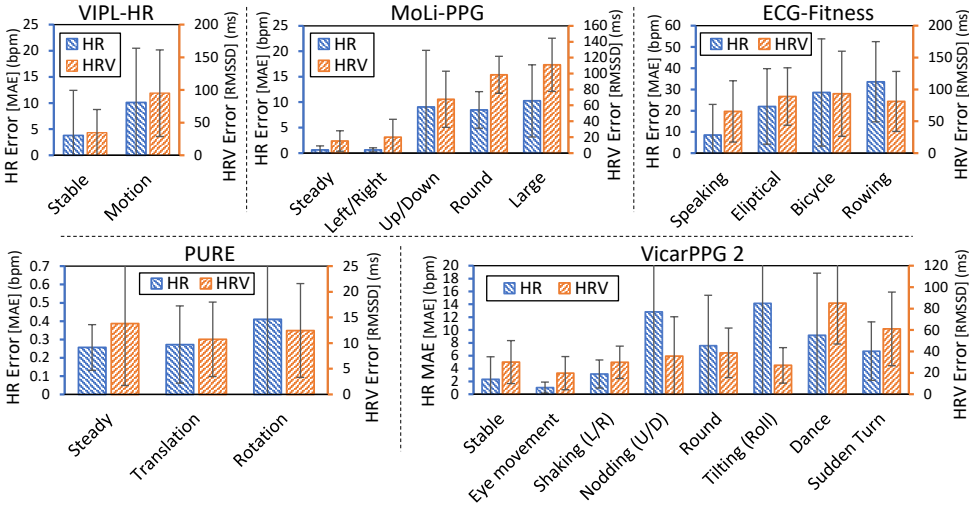


Figure 5.8: rPPG performance (MAE; error bars represent standard deviation) for different types of subject movements. Large/sudden multi-axis movement have the largest impact on accuracy. Interestingly, vertical head nodding motion seems to produce much higher error than horizontal head shaking motion of the same intensity.

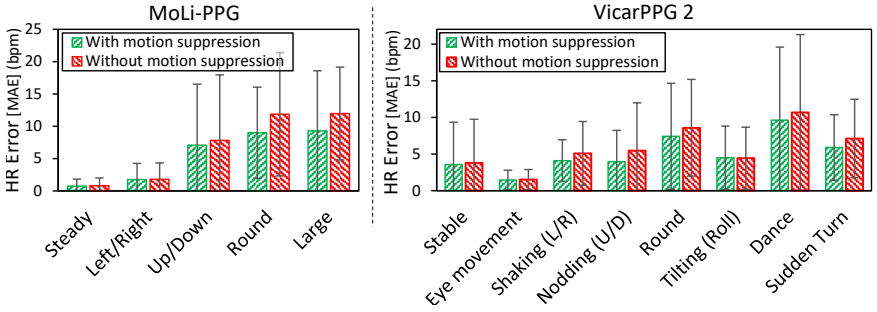
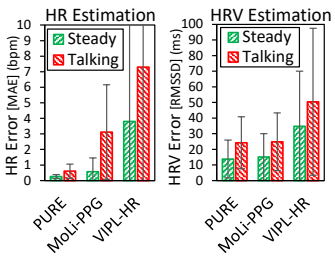
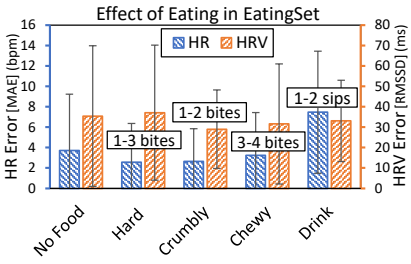


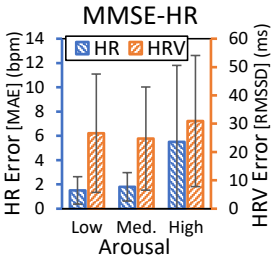
Figure 5.9: rPPG performance (MAE; error bars represent standard deviation) with and without the rhythmic motion noise suppression, for different types of subject movements on VicarPPG 2 and MoLi-PPG. In both datasets, motion noise suppression results in lower errors, especially when the subject movement is large.



(a) Effect of talking



(b) Effect of eating



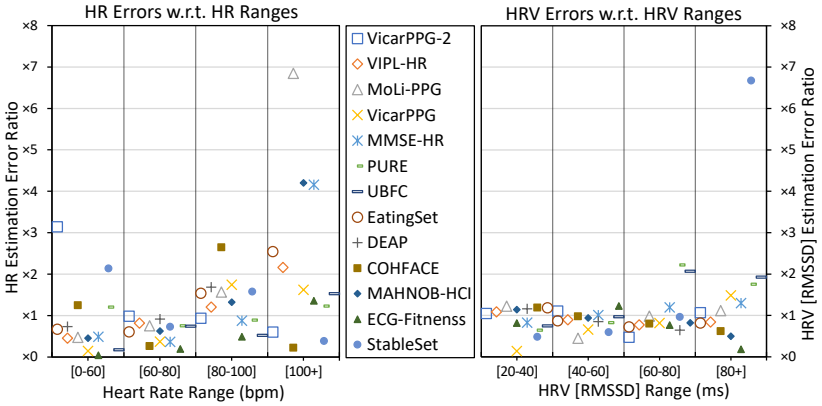
(c) Effect of facial expressions

Figure 5.10: rPPG performance (in terms of MAE; error bars represent standard deviation) somewhat degrades when subjects perform facial activity like taking (a) or expressing emotions (c). This can be partly attributed to non-trivial facial deformations causing changes in skin appearance. However, no significant performance degradation is noticed due to chewing motion of the mouth. Error while eating (b) is primarily caused by facial occlusion when subjects take a bite or a sip.

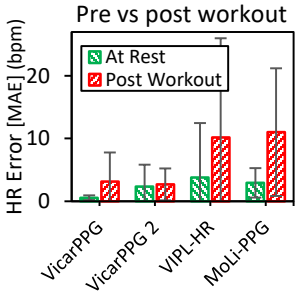
group the videos from all datasets based on their average ground truth heart rate / HRV and measure the rPPG performance on them separately. Figure 5.11a plots the relative error rate ratio (MAE) per group for each dataset w.r.t. the average error over the whole dataset. It can be noticed in Figure 5.11b that while the error rates in the post-workout condition does seem to be marginally higher than the baseline condition, the proposed method performs with sufficiently good accuracy in all conditions. On VicarPPG, closer examination revealed that the variable frame-rate of some videos often drops sharply. This affects the estimation of higher HRs more severely as the Nyquist sampling frequency requirement is also higher. More generally, heart rate estimation in higher HR ranges appears to produce higher errors more often, as seen in Figure 5.11a. A contributing factor for this could be the presence of higher frequency noise in the same frequency range as the higher heart rates. For HRV, no such trends could be observed in relation to the HRV ranges.

Influence of video compression

Pixel-level noise caused by common video compression formats can be a major source of errors for rPPG methods. In principle, the cleanest signal is obtained from uncompressed video frames, but this can be impractical due to their large size. To study this further, we evaluate our methods on a range of video compressions levels and formats to determine the trade-off between rPPG accuracy and video size/bitrate. The results can be seen in Figure 5.12 for the PURE dataset. The result show that among lossy encodings (denoted by circle), H.265 format best retains the rPPG pulsating component in the skin pixels of the video, resulting in high HR estimation accuracy, while maintaining the lowest bitrate.



(a) rPPG performance in HR and HRV ranges



(b) Post-workout conditions

Figure 5.11: rPPG performance for (a) HR/HRV ranges in terms of MAE ratio w.r.t the average mean absolute error per dataset; and (b) rested vs post-workout conditions in terms of MAE (error bars represent standard deviation). With a few exceptions, HR estimation in higher ranges (including during post-workout conditions) is more often less accurate than in lower ranges.

For example, accuracy only drops by ~0.3 bpm while the bitrate reduces by two orders of magnitude (26 Mbps to 0.26 Mbps). These results agree with the findings in [65]. Note that even under the lowest compression settings, these formats are not lossless and they result in a change in rPPG accuracy. Among the truly lossless codecs, FFV1 is able to encode the videos most efficiently. It results in zero drop in rPPG accuracy while reducing bitrates by almost an order of magnitude w.r.t raw videos. This can make rPPG dataset storage management much easier: for example, ECG-Fitness originally takes up 1.05 TB, but can be compressed with FFV1 to under 150 GB without losing any information.

Effect of camera type

A closely related factor that also affects rPPG analysis is the camera type, which determines the signal acquisition quality. Figure 5.13 provides a comparison of HR and HRV accuracy for different camera types used in VIPL-HR and MoLi-PPG datasets. As can be seen, the HD cameras provide the best performance in both datasets, likely due to their superior sensor, low internal compression, and higher frame-rate. However, we can also see that a modern webcam can closely match the performance of an HD camera under realistic condition (especially in VIPL-HR). The mobile front camera (Huawei P9) produces the

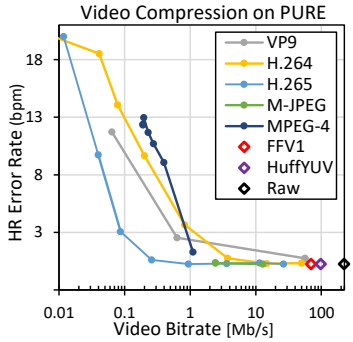


Figure 5.12: Effect of video compression on rPPG performance (MAE; PURE dataset). As videos are compressed to reduce their bitrate and storage size, rPPG accuracy decreases. H.265 offers the best trade-off between them among lossy encodings (denoted by circle). Among truly lossless encodings (denoted by diamond), FFV1 stores videos most efficently without affecting rPPG accuracy at all.

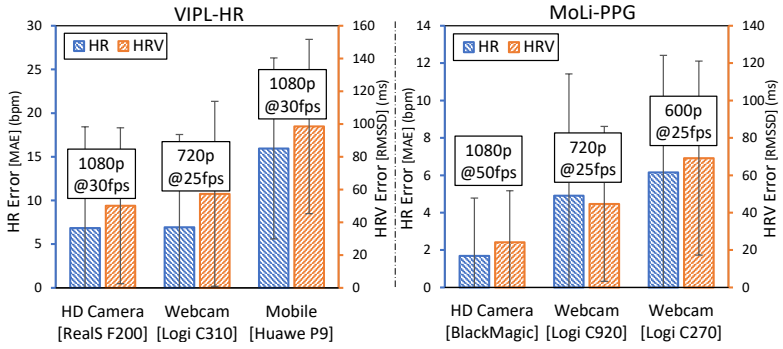


Figure 5.13: rPPG performance (MAE; error bars represent standard deviation) w.r.t camera type. HD cameras produce the best results likely due to their superior sensor. Modern webcams perform fairly well too.

worst performance in VIPL-HR, likely caused by it's significantly lower video acquisition quality.

Processing Speeds

Face Finding & Modelling	Skip Pixel Selection	rPPG Algorithm	Total	Frame Rate
31.89±17.2 ms	**0.43**±0.2 ms	**0.56**±0.2 ms	**32.88**±18.8 ms	~**30.4** fps

Table 5.4: The processing speed of individual components of the proposed method's pipeline and the total frame rate (640 × 480 pixel input) on an Intel Xeon E5 CPU. The main bottleneck is face finding and modelling, while the rest require negligible time.

For real time application, processing speed is just as vital as prediction accuracy. The average CPU processing times of our method and its individual components are listed in Table 5.4 (on an Intel Xeon E5 processor). The method performs the full analysis with a good real-time speed for a video resolution of 640×480 pixels. The only bottleneck is the face finding and modelling step, which is modular w.r.t the rPPG pipeline and can be swapped out for a faster implementation.

5.6 Discussion and Conclusions

We were able to obtain successful and promising results from our appearance modelling and signal-processing based rPPG method. The results show that this unsupervised method achieves high accuracies, matching or surpassing state-of-the-art on six public datasets: PURE, VIPL-HR, MoLi-PPG, VicarPPG, UBFC-RPPG, and MMSE-HR. In fact, the accuracy of our method for heart rate analysis is in the same range or beyond several fully supervised deep learning methods, albeit without any rPPG specific training or fine-tuning. We also surpass all existing methods for heart rate variability estimation and set some of the first benchmarks for heart rate variability analysis on these datasets.

Through an exhaustive 13-dataset evaluation (including release of a new public dataset: VicarPPG 2), the strengths and weaknesses of our method were highlighted. We showed that the proposed method handles most realistic variations in illumination, movement, and facial activity well. This can be attributed to the appearance modelling and noise suppression steps in the pipeline. However, certain combinations and extreme cases of these conditions proved challenging: overtly bright or flickering lighting and large head and body movements (e.g. during exercising). Our study provided some unique insights about the rPPG analysis in terms of performance while eating and emoting facial expressions: chewing motion during eating did not result in larger errors, while HR analysis during high facial arousal proved marginally challenging.

We also explicitly studied the impact of additional recording factors like video compression and camera type: H.265 and FFV1 emerged as clear winners in terms of preserving plethysmographic information in the skin pixels efficiently; higher quality rPPG signal can be obtained from HD cameras, but modern webcams also provide good results. High video compression noise was observed to be a clear limitation of our signal-processing method, especially in comparison with deep learning based method. Several deep learning methods have shown good results on such datasets, while they fail to match our method in cases with lower compression. This could be because the deep network is able to learn the spatial patterns of this compression noise and filter them out. In contrast, in lower compression cases, our prior domain knowledge assumptions perform more accurately. While this makes our method well suited for modern videos, deep learning might be better suited for processing archival videos, often subject to higher compression.

Finally, we demonstrated how the ground truth PPG and ECG signals provided with most datasets can be highly noisy, leading to incorrect peak detection and resulting in substantial miscalculation of heart rate and heart rate variability measures. To tackle this, we introduced the CleanerPPG set: a collection of hand-cleaned ground truth peaks for 13 major public datasets. Using this ground truth ensures a fairer and more accurate evaluation. This set is intended to continuously grow with community support.

References

[1] A. Gudi, M. Bittner, and J. van Gemert, *Real-time webcam heart-rate and variability estimation with clean ground truth for evaluation,* Applied Sciences **10**, 1 (2020).

[2] A. Gudi, M. Bittner, R. Lochmans, and J. van Gemert, *Efficient real-time camera based estimation of heart rate and its variability,* in *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019 (2019) pp. 1570–1579.

[3] J. Allen, *Photoplethysmography and its application in clinical physiological measurement,* Physiological measurement **28**, R1 (2007).

[4] M. Hassan, A. S. Malik, D. Fofi, N. Saad, B. Karasfi, Y. S. Ali, and F. Mériaudeau, *Heart rate estimation using facial video: A review,* Biomedical Signal Processing and Control **38**, 346 (2017).

[5] W. Verkruysse, L. O. Svaasand, and J. S. Nelson, *Remote plethysmographic imaging using ambient light.* Optics express **16**, 21434 (2008).

[6] W. Chen and D. McDuff, *Deepphys: Video-based physiological measurement using convolutional attention networks,* in *Proceedings of the European Conference on Computer Vision* (Springer, Cham, 2018) pp. 349–365.

[7] R. Spetlik, J. Cech, V. Franc, and J. Matas, *Visual heart rate estimation with convolutional neural network,* in *British Machine Vision Conference* (2018).

[8] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. T. Freeman, *Eulerian video magnification for revealing subtle changes in the world,* ACM Transactions on Graphics **31** (2012).

[9] Y. Zhang, S. L. Pintea, and J. C. Van Gemert, *Video acceleration magnification,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017) pp. 529–537.

[10] R. E. Kleiger, P. K. Stein, and J. T. Bigger, *Heart Rate Variability: Measurement and Clinical Utility,* Annals of Noninvasive Electrocardiology , 88 (2005).

[11] F. Shaffer and J. Ginsberg, *An overview of heart rate variability metrics and norms,* Frontiers in Public Health **5**, 258 (2017).

[12] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, *A multimodal database for affect recognition and implicit tagging,* IEEE transactions on affective computing **3**, 42 (2012).

[13] S. Koelstra, C. Muhl, M. Soleymani, J. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, *Deap: A database for emotion analysis ;using physiological signals,* IEEE Transactions on Affective Computing **3**, 18 (2012).

[14] P. V. Rouast, M. T. Adam, R. Chiong, D. Cornforth, and E. Lux, *Remote heart rate measurement using low-cost rgb face video: a technical literature review,* Frontiers of Computer Science **12**, 858 (2018).

[15] A. R. Pérez-Riera, R. Barbosa-Barros, R. Daminello-Raimundo, and L. C. de Abreu, *Main artifacts in electrocardiography,* Annals of Noninvasive Electrocardiology **23**, e12494 (2018).

[16] Y. Sun and N. Thakor, *Photoplethysmography revisited: from contact to noncontact, from point to imaging,* IEEE Transactions on Biomedical Engineering **63**, 463 (2015).

[17] D. McDuff, S. Gontarek, and R. W. Picard, *Improvements in remote cardiopulmonary measurement using a five band digital camera,* IEEE Transactions on Biomedical Engineering **61**, 2593 (2014).

[18] D. McDuff, S. Gontarek, and R. W. Picard, *Remote detection of photoplethysmographic systolic and diastolic peaks using a digital camera,* IEEE Transactions on Biomedical Engineering **61**, 2948 (2014).

[19] M.-Z. Poh, D. J. McDuff, and R. W. Picard, *Advancements in noncontact, multiparameter physiological measurements using a webcam,* IEEE transactions on Biomedical Engineering **58**, 7 (2010).

[20] G. De Haan and V. Jeanne, *Robust pulse rate from chrominance-based rppg,* IEEE Transactions on Biomedical Engineering **60**, 2878 (2013).

[21] X. Li, J. Chen, G. Zhao, and M. Pietikainen, *Remote heart rate measurement from face videos under realistic situations,* in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2014) pp. 4264–4271.

[22] H. E. Tasli, A. Gudi, and M. den Uyl, *Remote ppg based vital sign measurement using adaptive facial regions,* in *2014 IEEE International Conference on Image Processing (ICIP)* (IEEE, 2014) pp. 1410–1414.

[23] S. Tulyakov, X. Alameda-Pineda, E. Ricci, L. Yin, J. F. Cohn, and N. Sebe, *Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 2396–2404.

[24] W. Wang, A. C. den Brinker, S. Stuijk, and G. de Haan, *Algorithmic principles of remote ppg,* IEEE Transactions on Biomedical Engineering **64**, 1479 (2016).

[25] M. P. Tarvainen, P. O. Ranta-Aho, and P. A. Karjalainen, *An advanced detrending method with application to hrv analysis,* IEEE Transactions on Biomedical Engineering **49**, 172 (2002).

[26] Y. LeCun and Y. Bengio, *The handbook of brain theory and neural networks,* (MIT Press, 1998) Chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258.

[27] X. Niu, S. Shan, H. Han, and X. Chen, *Rhythmnet: End-to-end heart rate estimation from face via spatial-temporal representation,* IEEE Transactions on Image Processing **29**, 2409 (2019).

[28] Z. Yu, X. Li, X. Niu, J. Shi, and G. Zhao, *Autohr: A strong end-to-end baseline for remote heart rate measurement with neural searching,* IEEE Signal Processing Letters **27**, 1245 (2020).

[29] H. Van Kuilenburg, M. Wiering, and M. Den Uyl, *A model based method for automatic facial expression recognition,* in *European Conference on Machine Learning* (Springer, 2005) pp. 194–205.

[30] K. Alghoul, S. Alharthi, H. Al Osman, and A. El Saddik, *Heart rate variability extraction from videos signals: Ica vs. evm comparison,* IEEE Access **5**, 4711 (2017).

[31] Y. Sun, S. Hu, V. Azorin-Peris, R. Kalawsky, and S. E. Greenwald, *Noncontact imaging photoplethysmography to effectively access pulse rate variability,* Journal of biomedical optics **18**, 061205 (2012).

[32] A. M. Rodríguez and J. Ramos-Castro, *Video pulse rate variability analysis in stationary and motion conditions,* Biomedical engineering online **17**, 11 (2018).

[33] M. Finžgar and P. Podržaj, *Feasibility of assessing ultra-short-term pulse rate variability from video recordings,* PeerJ **8**, e8342 (2020).

[34] R. Stricker, S. Müller, and H.-M. Gross, *Non-contact video-based pulse rate measurement on a mobile service robot,* in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication* (IEEE, 2014) pp. 1056–1062.

[35] P. Li, Y. Benezeth, K. Nakamura, R. Gomez, C. Li, and F. Yang, *An improvement for video-based heart rate variability measurement,* in *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)* (IEEE, 2019) pp. 435–439.

[36] R. Song, H. Chen, J. Cheng, C. Li, Y. Liu, and X. Chen, *Pulsegan: Learning to generate realistic pulse waveforms in remote photoplethysmography,* arXiv preprint arXiv:2006.02699 (2020).

[37] S. Bobbia, R. Macwan, Y. Benezeth, A. Mansouri, and J. Dubois, *Unsupervised skin tissue segmentation for remote photoplethysmography,* Pattern Recognition Letters **124**, 82 (2019).

[38] R.-Y. Huang and L.-R. Dung, *Measurement of heart rate variability using off-the-shelf smart phones,* Biomedical engineering online **15**, 11 (2016).

[39] X. Niu, H. Han, S. Shan, and X. Chen, *VIPL-HR: A Multi-modal Database for Pulse Estimation from Less-Constrained Face Video,* in *Lecture Notes in Computer Science*, Vol. 11365 LNCS (2019) pp. 562–576.

[40] M. Artemyev, M. Churikova, M. Grinenko, and O. Perepelkina, *Robust algorithm for remote photoplethysmography in realistic conditions,* Digital Signal Processing , 102737 (2020).

[41] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features,* in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1 (2001).

[42] G. De Haan and A. Van Leest, *Improved motion robustness of remote-ppg by using the blood volume pulse signature,* Physiological measurement **35**, 1913 (2014).

[43] M. Malik, J. T. Bigger, A. J. Camm, R. E. Kleiger, A. Malliani, A. J. Moss, and P. J. Schwartz, *Heart rate variability: Standards of measurement, physiological interpretation, and clinical use,* European heart journal **17**, 354 (1996).

[44] P. Welch, *The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms,* IEEE Transactions on audio and electroacoustics **15**, 70 (1967).

[45] K. Oy, *About HRV* (2020), available online: `https://www.kubios.com/about-hrv` (accessed on May 5, 2020).

[46] Z. Zhang, J. M. Girard, Y. Wu, X. Zhang, P. Liu, U. Ciftci, S. Canavan, M. Reale, A. Horowitz, H. Yang, *et al.*, *Multimodal spontaneous emotion corpus for human behavior analysis,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 3438–3446.

[47] G. Heusch, A. Anjos, and S. Marcel, *A reproducible study on remote heart rate measurement,* arXiv preprint arXiv:1709.00962 (2017).

[48] T. B. Fitzpatrick, *The validity and practicality of sun-reactive skin types i through vi,* Archives of dermatology **124**, 869 (1988).

[49] C. M. MacLeod, *Half a century of research on the stroop effect: an integrative review.* Psychological bulletin **109**, 163 (1991).

[50] E. Billauer, *peakdet: Peak detection using MATLAB* (2012), available online: `http://billauer.co.il/peakdet.html` (accessed on March 13, 2020).

[51] H. Qi, Z. Guo, X. Chen, Z. Shen, and Z. J. Wang, *Video-based human heart rate measurement using joint blind source separation,* Biomedical Signal Processing and Control **31**, 309 (2017).

[52] R. Macwan, S. Bobbia, Y. Benezeth, J. Dubois, and A. Mansouri, *Periodic variance maximization using generalized eigenvalue decomposition applied to remote photoplethysmography estimation,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2018) pp. 1332–1340.

[53] M.-Z. Poh, D. J. McDuff, and R. W. Picard, *Non-contact, automated cardiac pulse measurements using video imaging and blind source separation.* Optics express **18**, 10762 (2010).

[54] H. Demirezen and C. E. Erdem, *Remote photoplethysmography using nonlinear mode decomposition,* in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, 2018) pp. 1060–1064.

[55] W. Wang, S. Stuijk, and G. De Haan, *A novel algorithm for remote photoplethysmography: Spatial subspace rotation,* IEEE transactions on Biomedical Engineering **63**, 1974 (2015).

[56] M. Finžgar and P. Podržaj, *A wavelet-based decomposition method for a robust extraction of pulse rate from video recordings,* PeerJ **6**, e5859 (2018).

[57] R. Macwan, Y. Benezeth, and A. Mansouri, *Heart rate estimation using remote photoplethysmography with multi-objective optimization,* Biomedical Signal Processing and Control **49**, 24 (2019).

[58] C. Zhao, C.-L. Lin, W. Chen, M.-K. Chen, and J. Wang, *Visual heart rate estimation and negative feedback control for fitness exercise,* Biomedical Signal Processing and Control **56**, 101680 (2020).

[59] R. Song, S. Zhang, J. Cheng, C. Li, and X. Chen, *New insights on super-high resolution for video-based heart rate estimation with a semi-blind source separation method,* Computers in biology and medicine **116**, 103535 (2020).

[60] M. Hu, D. Guo, X. Wang, P. Ge, and Q. Chu, *A novel spatial-temporal convolutional neural network for remote photoplethysmography,* in *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics* (IEEE, 2019) pp. 1–6.

[61] W. Sun, H. Wei, and X. Li, *No-contact heart rate monitoring based on channel attention convolution model,* in *Eleventh International Conference on Graphics and Image Processing*, Vol. 11373 (International Society for Optics and Photonics, 2020) p. 113732T.

[62] Y.-Y. Tsou, Y.-A. Lee, C.-T. Hsu, and S.-H. Chang, *Siamese-rppg network: remote photoplethysmography signal estimation from face videos,* in *Proceedings of the 35th Annual ACM Symposium on Applied Computing* (2020) pp. 2066–2073.

[63] Z. Yu, X. Li, and G. Zhao, *Remote photoplethysmograph signal measurement from facial videos using spatio-temporal networks,* in *30th British Machine Vision Conference 2019* (2019) p. 277.

[64] M. Den Uyl and H. Van Kuilenburg, *The facereader: Online facial expression recognition,* in *Proceedings of measuring behavior*, Vol. 30 (Citeseer, 2005) pp. 589–590.

[65] C. Zhao, C.-L. Lin, W. Chen, and Z. Li, *A novel framework for remote photoplethysmography pulse extraction on compressed videos,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2018) pp. 1299–1308.

6

Discussion

This dissertation studies techniques for reducing the need for resources in machine vision systems through incorporation of prior knowledge.

In **Chapter 2**, for the task of localizing objects in images using weak supervision, prior knowledge about the distribution of objects' bounding boxes is exploited. This leads to a novel object-extent pooling technique. Together with a hand-crafted projection algorithm, this results in a supervision-efficient detector with low computational costs.

Chapter 3 questions the role of full-face spatial context for the primarily local task of gaze estimation from images. It demonstrates how processing additional context in the input does not yield meaningful performance gains, yet it adds to the computational requirements. This chapter also examines data efficiency of different calibration techniques for gaze projection imparted with varying amounts of prior knowledge.

Chapter 4 attempts to exploit the spatial contextual structures in images by suggesting a proximally sensitive distance function based on a Gaussian kernel. Results demonstrate the application of such a function for tasks requiring high data-efficiency such as few-shot anomaly and facial occlusion detection. Results also hint towards the suitability of such local sensitivity in loss functions when available training data or model complexity is low.

Finally, **Chapter 5** demonstrates how an unsupervised prior-knowledge based method can be better than (deep) learning models for the task of vital signs estimation from faces: both in terms of accuracy, as well as informational & computational efficiency. With its detailed study of affecting factors, the chapter also provides insights into the validity of prior-knowledge assumptions in the face of video compression noise and task complexity.

Several common themes emerge in the observations from these chapters. These themes are discussed in the following sections.

6.1 Prior Knowledge and Efficiency

Incorporation of human learning Prior knowledge about a particular task developed by the human expert is based on her/his experience with the task. Humans are known to be good few-shot learners [1] and can generalize about a task from relatively few examples. Therefore, incorporation of prior knowledge into the model can be seen as assimilating the findings of humans' few-shot learning capability into the system.

For example, in Chapter 2, since the human designer already understands the correlation between the activation and object location extent (by observation), this can be 'hard-coded' into the model. This is realized in the form of the *SPAM* pooling layer. Similarly, the geometric and hybrid calibration methods studied in Chapter 3 follow the same principle: since the geometric relations between gaze angles and gaze points are already known, this can be programmed into the system. The introduction of the fixed gaussian kernel in the proximally sensitive error function in Chapter 4, and the design of wide/narrow band frequency filter in Chapter 5 are other examples of this. These instances show how understanding of the underlying mechanism learnt by humans can be valuable in designing informationally efficient systems. Therefore, a greater focus must be placed on the study of the underlying fundamental processes at work in various computer vision tasks, rather than on just optimizing the learning algorithm/model.

Training data and learning capacity Given sufficient input information and no restrictions on training data/supervision, a capable machine learning model could learn the relevant associations required for performing a task directly from the data. The value of prior knowledge shines when the availability of training data/supervision is scarce, or when the model's learning capacity is low. Because the model has limited opportunity to learn by itself, pre-programmed prior knowledge mechanisms incorporated in the system become more beneficial.

In Chapter 2, this phenomenon might be attributed to the strong performance of the hand-crafted SPAM pooling layer, where location labels were not available for training. Given full supervision, the model *can* learn by itself to associate maximum activations in the feature maps with the location/extent of the object. However, since only weak-supervision is available, the model is given this knowledge *a priori* by means of hand-crafting the pooling layer; and the model benefits from it.

This observation is more explicitly seen in Chapter 3: when the number of calibration data points are fewer, geometric and hybrid calibration methods (which use prior knowledge) perform better than machine learning calibration. However, when the data points become abundant, machine learning surpasses geometric calibration. Quite similar observation is also partially seen in the results of Chapter 4: the gap in accuracy between using *PSE* (with gaussian kernel) over *MSE* as loss functions is largest when the amount of training data or the model complexity is lowest, and this gap reduces when these parameters are increased. These examples demonstrate some of the suitable scenarios where prior knowledge can overcome the limitations in informational and computational resources. These scenarios motivate against abandoning research into "hand-crafting" computer vision solutions in favour of only training learning-based models.

(Deep) Learning is not always better While learning based systems typically perform well with sufficient training data, there do exists certain image processing tasks for which prior knowledge based system can be better suited than learning based system even when data is abundant. These are tasks for which the underlying mechanism is well understood and implementable.

An example of this is seen in the comparison of hybrid and machine learning calibration methods in Chapter 3. The purely data-driven machine learning method is never able to match/surpass the accuracy of the prior-knowledge based hybrid method. This is likely because the implemented geometrical/mathematical relations between the input and output are straightforward and well established. Another example is in Chapter 5: the proposed hand-crafted unsupervised *rPPG* pipeline estimates vital signs more accurately (and faster) than several fully supervised deep learning estimators. This is likely because the underlying mechanism behind estimating vital signs from videos (skin pixel tracking) is well known and can be squarely implemented as prior knowledge. Conversely, while end-to-end trained deep neural networks also are able to learn an approximation of it from the given data, this is not precise enough. These examples demonstrate scenarios where data driven models can be ill-suited even in the abundance of data.

At the same time, if the complexity of the task increases and requires modelling of higher level interactions with unclear mechanisms, the assumptions in the prior knowledge could be incorrect. Consequently, incorporation of such prior knowledge can limit the generalizability of hand-crafted methods. For example, a failure point of the hand-crafted *rPPG* method in Chapter 5 becomes apparent when video compression noise is introduced. Deep learning methods appear to overcome this, possibly by learning to recognize and filter out the spatial patters of the compression noise. The mechanism surrounding such noise is not trivial, and therefore hand-crafting such a filter is not feasible. In these situations with non-trivial distractors, learning-based models seem better suited than hand-crafted ones.

These sets of observations motive further research into understanding the complexity/simplicity of various computer vision tasks. Such studies can provide insight into the suitability of different categories of techniques for a given task. This can help avoid "over-engineering" solutions for tasks that can be solved by simpler approaches.

6.2 Context and Efficiency

Context is beneficial Often context can help the model achieve better results by exploiting patterns in the scene, such as locations of objects in an image: e.g., ceiling fans are likely on the ceiling. This can especially be useful for tasks that are subjective in nature and require placing judgements in a global understanding of the scene. Such tasks may including image style transfer, scene understanding for image to text translation, detection of unclear/ambiguous objects in scene, etc. In these cases, adding context almost always leads to better accuracy, albeit at the cost of efficiency.

Distractions and biases However, overexploiting this context can also be harmful and lead to edge case failures: e.g., uninstalled ceiling fans can also be placed on the floor. Context overexploitation is essentially taking advantage of biases in the dataset, which can result in better reported accuracy. However, real-world application can have unseen

scenarios where such a model can fail. For example, for the task of facial expression classification, a model analyzing a colour image of a face might overexploit the skin colour information to make its prediction due to potential biases in the training dataset. This issue can be somewhat rectified by pruning the colour context information out of the input and training the model on grayscale images, which forces the model to focus on other features. Thus, limiting contextual information can potentially simplify the task for the model and reduce overfitting/biases.

Computational costs Another drawback of processing excessive context in the input is the computational costs associated with it. In this thesis, Chapter 3 studies the role of contextual input information for the task of gaze tracking from webcams. While several work have shown good performance using full face as input [2–4], the results from this chapter demonstrated how similar performance can be achieved by using single eye crops instead, which require a fraction of computational resources. This is also seen in Chapter 5, where the hand-crafted rPPG pipeline relies on pre-computed spatial averages of skin pixel values. This contributes to making the core algorithm run with beyond-real-time speeds. In contrast, majority of the supervised deep learning methods rely on analysing all the pixels in the face region, which can be several orders of magnitude larger, thereby requiring more processing. Therefore, contextual information has computational costs, and this must be taken into account when considering any potential accuracy benefits.

Informational costs A pitfall of training models with rich contextual information is that it raises the informational needs of the model. Data collection, storage, and labelling requirements can all increase if additional contextual information needs to be captured, saved or annotated. Further, to ensure that a model does not overfit on the contextual information in the input, the training data must ensure good diversity of this context, which further increases collection effort. For example, for the gaze estimation model form Chapter 3, it is likely that a large quantity of training data is required if full-face images are used, so as to avoid overfitting on non-eye facial features. In comparison, it might be possible to train models with much less data if only the smaller eye-crops are used. This is a research direction that beckons further investigation.

6.3 Developmental Effort

Finally, based on the experiences from this dissertation, it can be said that manual incorporation of prior-knowledge and hand-crafting is a tedious task requiring significant manual effort with prototyping and experimentation. This is especially true when comparing with end-to-end training of (deep) learning models based off of standard architectures /'foundation models' [5]. While experimentation and prototyping is still required for optimizing the performance of learning methods (model architecture, learning strategy, etc.), this is typically less effort than implementing non-standard pipelines designed from scratch.

Domain expertise One of the advantages of developing learning based systems is that domain expertise of the task is not required. Expertise in the learning mechanism is typically sufficient, and this can in theory be applied to any task without necessarily

requiring complete understand of the task domain. In contrast, such task domain expertise is usually critical for the design of hand-crafted systems. In fact, this expertise is the basis of prior knowledge. This can be seen as a disadvantage of hand-crafted systems.

A prime example of this disadvantage is the design of the prior-knowledge driven hand-crafted rPPG algorithm in Chapter 5. The knowledge of the underlying principle of remote photo-plethysmography was absolutely critical in making the design decisions. Such expertise is also the basis of the design of the geometric and hybrid calibration techniques introduced in Chapter 3: the 3D geometry and all mathematical relations need to be known for them to be implemented. The design of the object-extent *SPAM* pooling layer and the back-projection algorithm in Chapter 2, and the formulation of the proximally sensitive error in Chapter 4 also fall under this category. Therefore, developing hand-crafted methods must not be considered as a stand-in replacement for training learning-based methods.

6.4 Final Word

In the face of ever increasing amount of the informational availability and computational power, efficiency in machine vision systems still matters. Development of practical real-world applications of computer vision methods is constrained by several computational and informational limitations, where only well-efficient systems can operate adequately. The dazzle of high accuracy numbers can sometimes overshadow the costs being incurred to achieve them, and this dissertation attempts to shine the spotlight on these costs.

References

[1] B. M. Lake, T. Linzen, and M. Baroni, *Human few-shot learning of compositional instructions,* arXiv preprint arXiv:1901.04587 (2019).

[2] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, *Eye tracking for everyone,* in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[3] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, *It's written all over your face: Full-face appearance-based gaze estimation,* in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2017) pp. 2299–2308.

[4] C. Palmero, J. Selva, M. A. Bagheri, and S. Escalera, *Recurrent CNN for 3d gaze estimation using appearance and shape cues,* in *British Machine Vision Conference (BMVC)* (2018).

[5] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et al.*, *On the opportunities and risks of foundation models,* arXiv preprint arXiv:2108.07258 (2021).

Curriculum Vitæ

Amogh Anirudh Gudi

1989/09/19 Born in Udhampur, India.

Education

2005–2007 Senior High School
Army Public School (Dhaula Kuan), New Delhi

2007–2011 Bachelor of Engineering in Electronics and Instrumentation
Visvesvaraya Technological University (Dr. AIT), Bangalore

2012–2014 Master of Science in Artificial Intelligence
University of Amsterdam, Amsterdam

2016–2022 Ph.D. in Computer Science
Delft University of Technology, Delft
Thesis: Less Machine (=) More Vision
Promotor: Prof. dr. ir. M.J.T. Reinders

Work Experience

2014–2022* Machine Vision Research Engineer
Vicarious Perception Technologies (VicarVision), Amsterdam

List of Publications

13. **Amogh Gudi***, Marian Bittner*, Jan van Gemert, *Real-Time Webcam Heart-Rate and Variability Estimation with Clean Ground Truth for Evaluation*, Applied Sciences, Special Issue: Video Analysis for Health Monitoring, 2020.

12. **Amogh Gudi**, Xin Li, Jan van Gemert, *Efficiency in Real-Time Webcam Gaze Tracking*, IEEE/CVF International Conference on Computer Vision Workshops: Gaze Estimation and Prediction in the Wild, Glasgow (Virtual), 2020.

11. **Amogh Gudi***, Marian Bittner*, Roelof Lochmans, Jan van Gemert, *Efficient real-time camera based estimation of heart rate and its variability*, IEEE/CVF International Conference on Computer Vision Workshops: Computer Vision for Physiological Measurement, Seoul, 2019.

10. **Amogh Gudi**, Fritjof Büttner, Jan van Gemert, *Proximally Sensitive Error for Anomaly Detection and Feature Learning*, arXiv:2206.00506, Extended abstract, ICT.OPEN, Hilversum, 2019.

9. **Amogh Gudi***, Nicolai van Rosmalen*, Marco Loog, Jan van Gemert, *Object-Extent Pooling for Weakly Supervised Single-Shot Localization*, British Machine Vision Conference, London, 2017.

8. H. Emrah Tasli, **Amogh Gudi**, Paul Ivan, Marten den Uyl, *A method for stabilizing vital sign measurements using parametric facial appearance models via remote sensors*, Patent EP2960862B1, 22, 2017.

7. Agne Grinciunaite, **Amogh Gudi**, H. Emrah Tasli, Marten Den Uyl, *Human Pose Estimation in Space and Time using 3D CNN*, 14th European Conference on Computer Vision Workshops: Brave new ideas for motion representation in videos, Amsterdam, 2016.

6. **Amogh Gudi**, H. Emrah Tasli, Tim M. den Uyl, Andreas Maroulis, *Deep Learning based FACS Action Unit Occurrence and Intensity Estimation*, 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Ljubljana, 2015.

5. **Amogh Gudi**, H. Emrah Tasli, Max Welling, *Recognizing semantic features in faces using deep learning*, Master's thesis, University of Amsterdam, arXiv preprint arXiv:1512.00743, 2015.

4. H. Emrah Tasli, **Amogh Gudi**, Marten den Uyl, *Integrating Remote PPG in Facial Expression Analysis Framework*, 16th International Conference on Multimodal Interaction, Istanbul, 2014.

3. H. Emrah Tasli, **Amogh Gudi**, Marten den Uyl, *Remote PPG based Vital Sign Measurement using Adaptive Facial Regions*, IEEE International Conference on Image Processing, Paris, 2014.

2. **Amogh Gudi***, Patrick de Kok*, Georgios K Methenitis*, Nikolaas Steenbergen*, *Feature detection and localization for the RoboCup Soccer SPL*, Report, University of Amsterdam, 2013.

1. **Amogh Gudi**, M. Prasanna Kumar, *Automated Vehicle Verification System*, Proceedings of the 5th National Conference on Computing for Nation Development INDIACom, New Delhi, 2011.

Included in this thesis. Best paper or poster award. Registered patent.

* These authors contributed equally.

ISBN 978-94-6366-602-2