



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Doctor of Philosophy

**A High-efficiency Real-time Facial Emotion
Recognizer Using Deep Learning Architectures**

The Graduate School

of the University of Ulsan

Department of Electrical, Electronic and Computer Engineering

Muhamad Dwisnanto Putro

A High-efficiency Real-time Facial Emotion Recognizer
Using Deep Learning Architectures

Supervisor: Kang-Hyun Jo

A Dissertation

Submitted to
the Graduate School of the University of Ulsan
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Muhamad Dwisnanto Putro

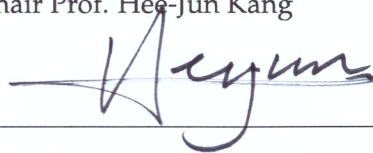
Department of Electrical, Electronic and Computer Engineering
University of Ulsan

May 2022


**A High-efficiency Real-time Facial Emotion Recognizer
Using Deep Learning Architectures**

This certifies that the dissertation
of Muhamad Dwisnanto Putro is approved:

Committee Chair Prof. Hee-Jun Kang



Committee Member and Supervisor Prof. Kang-Hyun Jo



Committee Member Prof. Young-Soo Suh



Committee Member Prof. Hyun-Deok Kang



Committee Member Prof. Jang-Sik Park



Department of Electrical, Electronic and Computer Engineering

University of Ulsan, South Korea

May 2022

“Dream big, and make it come true with action and persistence.”

UNIVERSITY OF ULSAN

ABSTRACT

Graduate School of Electrical Engineering

Department of Electrical, Electronic and Computer Engineering

Doctor of Philosophy

A High-efficiency Real-time Facial Emotion Recognizer Using Deep Learning Architectures

by Muhamad Dwisnanto Putro

Facial emotion recognition is a method to localize and predict human facial expressions. It identifies the texture of face elements in an interaction process. Besides, this domain is nonverbal communication that conveys facial indications to show feelings. Facial expression recognition work is a trending study field in human-robot interaction. This research plays an important role in supporting assistive robot performance. A real-time system is required to increase the robot's abilities and prevent misunderstandings caused by dynamic personal activities. Moreover, a practical application requests real-time performance from the computer vision technique on low-cost computing devices.

This work in this manuscript focuses on human facial emotion recognition in a real-world scenario that estimates the location of faces and fast identifies their expressions. The network efficiency does not ignore the predicted performance of each module. Therefore, the research in this thesis proposes high performance and efficiency using deep learning architecture for detecting face area and classifying the emotion from a live streams video. Each module works separately and operates smoothly by achieving real-time speed.

A complete real-time facial emotion recognizer consists of two-stage CNN-based architecture containing a face detector and a facial expression classification. The proposed face detection plays an essential role in filtering the face area from the background. It also avoids the prediction error of the single-label classification system when there is more than one face in an image. It utilizes several shallow layers

of convolution that form a lightweight architecture implemented as a real-time detector. However, this does not neglect its precision for localizing faces of varying sizes and poses. The proposed face detector contains two main parts, a backbone to discriminate specific components and multi-level detection to estimate the multiple-scale faces location. It also utilizes several techniques to improve the training performance, such as balanced loss function and tweaking of parameters configuration.

The facial expression classification module categorizes seven fundamental human emotions: neutral, fear, surprise, disgust, sad, happy, and anger. This system also focuses on the efficiency of computational and parameters to support lightweight and fast integrated systems. An efficient facial expression framework proposes a sequential attention network to enhance the backbone performance. It includes three modules, global attention to highlight the global context of features, channel attention, and dimension attention, which concentrate on the relationship of local elements in the channel and spatial dimension. Besides, It offers the Efficient Partial Transfer (EPT) module as an efficient extractor of facial features from an image. Augmentation of various facial poses increases reliability and capability to recognize non-frontal facial expressions. It supports the proposed system's performance, enabling implementation in a real-world scenario.

Several experimental results for each module show satisfying performance to each benchmark dataset and achieve competitive accuracy from competitors. It is due to the proposed modules that increase performance without producing redundant computing and parameters. The light network can precisely learn the characteristics of specific and global features in the data variation. Additionally, system integration demonstrates that the emotion recognizer operated at real-time speed on the CPU-based devices and an edge device.

Acknowledgements

I would like to express my gratitude for a special one, ALLAH SWT. My grateful praise is for God and Nourisher of the worlds. Thanks to my wife and parents, who always stayed beside me while studying in Korea. Particular gratitude to my advisor, Professor Kang-Hyun Jo. He authorized me to study under his guidance, encouragement, and support for all activities in South Korea. He always motivates me to develop continuously for future success. I would also thank the thesis committee: Prof. Hee-Jun Kang, Prof. Young-Soo Suh, Prof. Hyun-Deok Kang, and Prof. Jangsik Park, for their valuable comments and advice to improve this thesis.

I am grateful to the Intelligent Systems Laboratory fellows for sharing knowledge and meaningful discussion that can help me improve my understanding of computer vision and deep learning. All criticisms and suggestions encourage me to become a better person. Many thanks to the University of Ulsan and the Brain Korea Scholarship program that supports my study and research to produce works and contributions during the study process.

Contents

Abstract	iii
Acknowledgements	v
1 Introduction	1
1.1 Motivation and Background	1
1.2 Problem Description and Objective	3
1.3 Contributions	5
1.4 Disposition	6
2 Literature Review	7
2.1 Vision System	7
2.2 Image Classification	8
2.3 Object Detection	9
2.4 Classical Face Detector	9
2.5 Deep learning-based Face Detector	11
2.6 Facial Expression Network	14
2.7 Attention Mechanism	17
3 Face Detection Network	20
3.1 System Architecture	21
3.1.1 Backbone Module	22

3.1.2	Detection Module	24
3.1.3	Anchor Strategy	25
3.1.4	Balanced Loss Functions	26
3.2	Training Dataset and Implementation Details	27
3.3	Experiments and Results	28
3.3.1	Model Analysis	28
3.3.2	Comparison with other detectors	29
4	Facial Expression Network	35
4.1	Backbone module	36
4.1.1	Sequential Attention Module	37
4.2	Classifier Module	41
4.3	Implementation Setup and Dataset Configuration	43
4.4	Experimental Results	44
4.4.1	Backbone Analysis	44
4.4.2	Proposed Model Analysis	45
4.4.3	Evaluation on Dataset	48
5	Integrated Facial Emotion Recognition	51
5.1	Runtime Efficiency of Face Detector	52
5.2	Real-time Application of Face Detector on Low-cost Devices	54
5.3	Runtime Efficiency of Facial Emotion Recognizer	56
5.4	Real-time Application of Face Emotion Recognizer on Low-cost Devices	57
6	Conclusion	61
6.1	Conclusions	61
6.2	Future Works	63

A Publications	65
A.1 Journal	65
A.2 Conference	66
Bibliography	69

List of Figures

1.1	Illustration of a real-time face emotion recognizer.	2
1.2	Major components in the two-stage real-time face emotion recognition.	4
2.1	The process flow of human vision (Elgendy, 2020).	7
2.2	The parts of computer vision utilize sensing and an interpreting device (Elgendy, 2020).	8
2.3	An object detector with the sliding window strategy (Kurnianggoro, 2019).	9
2.4	A S3FD architecture (Zhang et al., 2017).	12
2.5	A TinaFace architecture (Zhu et al., 2020).	13
2.6	A YOLOV5Face architecture (Qi et al., 2021).	14
2.7	The ExpNet approach (Otberdout et al., 2020).	15
2.8	Overview of the ensemble of MLCNNs approach (Nguyen et al., 2019).	16
2.9	The CNN with visual attention (Sun, Zhao, and Jin, 2018).	17
2.10	The architecture of attention mechanism-based CNN network (Li et al., 2020).	19
3.1	Architecture of the face detector network.	22
3.2	Mini-inception module for face detector network.	23
3.3	Transition block for face detector network.	24
3.4	A approach of anchor density with 32×32 (a), 64×64 (b), and 96×96 (c) dimension.	25

3.5	Visualization of the detection results on AFW (a), PASCAL face (b), FDDB (c), and WIDER FACE (d).	30
3.6	Comparison of Average Precision (AP) on the AFW dataset.	31
3.7	Comparison of Average Precision (AP) on the PASCAL face dataset.	31
3.8	Comparison of true positive rate uses discrete ROC (Receiver Operat- ing Characteristics) curves on the FDDB dataset.	32
3.9	Comparison of Average Precision (AP) on the WIDER FACE dataset.	33
4.1	The general architecture of facial expression network.	35
4.2	Architecture of efficient backbone with transferred partial feature.	36
4.3	Global attention module.	38
4.4	Channel attention module.	39
4.5	Dimension attention module.	40
4.6	The modified Spatial Pyramid Pooling (SPP).	42
4.7	The heatmap visualizes the feature attention at each representation module.	46
4.8	A comprehensive evaluation of the proposed model in confusion ma- trix on (a) FER-2013 and (b) CK+ datasets.	48
4.9	A comprehensive evaluation of the proposed model in confusion ma- trix on (a) JAFFE and (b) KDEF datasets.	49
4.10	Confusion matrix of multi-pose evaluation on KDEF datasets.	50
5.1	Integration system of facial emotion recognizer.	51
5.2	The proposed model speed is compared with FaceBoxes at different input resolutions.	53
5.3	Visualization of the face detection results on live streams video at VGA (a), HD (b), and Full HD (c) resolution.	55
5.4	Visualization of detection results in the integrated system on a live streams video.	58

5.5 Comparison of integrated model speed implemented on low-cost devices. 59

List of Tables

3.1	Model analysis of each module.	28
4.1	Comparison of efficient backbone with other networks.	44
4.2	Ablative study of transferred partial module.	45
4.3	Model analysis of each proposed module.	45
4.4	Formation analysis of proposed attention modules.	46
4.5	Comparison of the proposed model with to other methods on different datasets.	47
4.6	Evaluation of multi-pose facial expression recognition on KDEF dataset.	50
5.1	Runtime efficiency compared to different face detectors on CPU.	52
5.2	Comparison of model speed in real-time application on the low-cost and an edge device	53
5.3	Comparison of the integrated model speed on Intel Core i5 CPU.	56

List of Abbreviations

AFW	A nnotated F aces in the W ild
AP	A verage P recision
CNN	C onvolution N eural N etwork
CPU	C entral P rocessing U nit
CR	C hannel R epresentation
DCNN	D eep C onvolution N eural N etwork
DR	D imension R epresentation
EPT	E fficient P artial T ransfer
GFLOP	G iga F loating-point O peration
GR	G lobal R epresentation
FDDB	F ace D etection D ata sets and B enchmarks
FPN	F eature P yramid N etwork
FPS	F rame P er S econd
GPU	G raphic P rocessing U nit
IoU	I ntersection O ver U nion
SGD	S tochastic G radient D escent
SSD	S ingle S hot D etector
SVM	S upport V ector M achine
TPR	T rue p ositive R ate
ReLU	R ectified L inear U nit
RGB	R ed G reen B lue
ROI	R egion O f I nterest
VGA	V ideo G raphic A rray
VGG	V isual G eometry G roup
WIDER	W eb I mage D ataset for E vent R ecognition

List of Symbols

p	label class
reg_i^{pred}	coordinates and size of predicted box
reg_i^{gt}	ground truth box
cls_i^{gt}	ground truth label class
cls_i^{pred}	predicted label class
Pos	number of positive boxes
L_{cls}	classification loss
m_i^{pred}	matching logic
cls_i^0	confidence score of non-objects
L_{reg}	regression loss
x_i	input features map
C	convolution layer
x^s	splited features map
Fl	flatten
$Pool$	pooling operation
E_x	excitation modules
G_x	global context
Tx_i	contextual information
x_i	the reshaped map
W	weight tensor
GP	global average pooling
GA	global representation
DA	dimension representation
CA	channel representation
t_j	transposed tensor
t_i	reshaped tensor
h_i	a valuable feature from channel representation

p_i	a valuable feature from dimension representation
δ	first balancing parameter
η	second balancing parameter
Γ	ReLU activation
\mathbb{N}	layer normalization

*Dedicated to
anyone who wants to learn*

Chapter 1

Introduction

1.1 Motivation and Background

Facial expressions, voice tone, and body gestures are communication approaches to identify human emotions. This research topic is a hot issue in computer vision (Xi et al., 2021) and plays an important position in Human-robot Interaction (HRI). It has the main task of connecting and synchronizing information between robots and users. Therefore, the misunderstanding of perception will impact the mistake of robot action and incompatibility with the purpose. Currently, robots are developing to collaborate with users, especially an assistive robot that needs social interaction techniques (Putro and Jo, 2018). It aims to help the daily activities of humans, even working all the time without compromising.

The proportion of interaction activities is 7% of the affective information is conveyed through words, 38% is delivered by tone, and 55% through facial expressions (Rawal and Stock-Homburg, 2021). This fact shows that facial emotions are needed to support affective communication. It encourages a social robot to comprehend human emotions correctly. On the other hand, HRI needs this robot to cooperate with users. The emotional information has to be fast identified by a robot. Thus, the approach is required to operate at a real-time speed so that robots can assist human activities efficiently. It is also driven by practical applications that demand computer vision methods to work quickly on low computing devices and adapt in real case scenarios. Figure 1.1 shows the illustration of a real-time facial emotion recognizer for human-robot interaction.



FIGURE 1.1: Illustration of a real-time face emotion recognizer.

Facial expression recognition is a classification method that predicts facial emotion categories by identifying critical facial features. Paul Ekman has described the six basic emotions: angry, disgust, happiness, surprise, sadness, and afraid (Ekman, 1992). Each facial expression contains different characteristics and specific correlations of features between the elements (Li and Deng, 2020). Eyes, cheeks, nose, eyebrows, lips, forehead, and mouth are interesting facial features and have a crucial influence on predicting expression (Kumari, Rajesh, and Pooja, 2015). In addition, the shape and texture of these features show different characteristics in each gesture. This composition is informative knowledge for the final decision of the prediction.

Previous works have successfully predicted facial expression based on the conventional method (Hu et al., 2019; He and Chen, 2020). Hu et al. have used local descriptors to obtain specific facial components. Center-Symmetric Local Signal Magnitude Pattern (CS-LSMP) filters facial texture from an input image. It captures the difference of gray information from neighbor pixels and represents it in magnitude information. In addition, He et al. have applied a traditional extractor to recognize person-independent for recognizing the expressions. It improves the LBP (Local Binary Pattern) and singular value decomposition methods. Conventional feature extraction methods are weak for discriminating facial features, so their works generate a large number of false predictions.

Image classification works already implemented a Convolutional Neural Network (CNN) as a robust feature extractor (Zeiler and Fergus, 2014; Hossain, Al-Hammadi, and Muhammad, 2019; Shahbaz and Jo, 2020). The convolutional operation utilizes updated filters to discriminate the essential features of an object. Then, it employs back-propagation to update those weights. This method has been implemented in several facial expression tasks and successfully performed with high accuracy. The Deep Convolutional Neural Network (DCNN) has been utilized as an extractor to deliver satisfying performance (Hayale, Negi, and Mahoor, 2021; Otberdout et al., 2020). However, those architectures produce a lot of parameters and heavy computations. The CNN model requires high GPU usage to work quickly, while this accelerator is not cheap. Meanwhile, a practical application demands a simulation with real-time performance and can operate on low-cost devices. A DCNN approach tends to run slowly on these devices.

Although a DCNN architecture provides powerful performance, this model requires high processing times and heavy computations. This slow computation restricts the applicability of deep learning models. Even though the shallow model allows fast operation on low-cost devices, the deeper model delivers better performance and demands more computing resources (Kim et al., 2019). Therefore, a balance of trade-off between performance and efficiency is needed to support the implementation of robotics.

1.2 Problem Description and Objective

In general, the image classification system predicts a specific class. It also finds the important features and the relationship between elements. However, this system is constrained by accuracy when it contains complex background components that obtain a lot of false predictions. The single category classification has significant error occurs when there is more than one face. In order to address this problem, the proposed system localizes the facial area at the beginning of the stage to filter it from the background elements that produce facial patches. Then, it applies a facial expression classification system to each patch. We offer an integrated deep learning model that combines face detection and facial expression classification to recognize facial emotions accurately. In addition, the proposed system emphasizes increasing

the speed of data processing to be efficiently implemented on low-cost devices.

An integrated face emotion recognizer contains two main parts, as described in Figure 1.2. An input image is processed by face detection to extract the region of interest (RoI). This approach avoids background noise helping the facial emotion classifier to focus on predicting only the facial area. Efficient face detectors are designed to support a real-time system that works on CPU and edge devices. Although the shallow architecture has light computation, the proposed face detector overcomes problems with variations in scale, pose, occlusion, and extreme backgrounds. The second stage is facial expression classification employs an efficient backbone layer and the robust attention module to extract specific and global essential features. The network's end predicts seven expressions that represent basic human facial emotions.

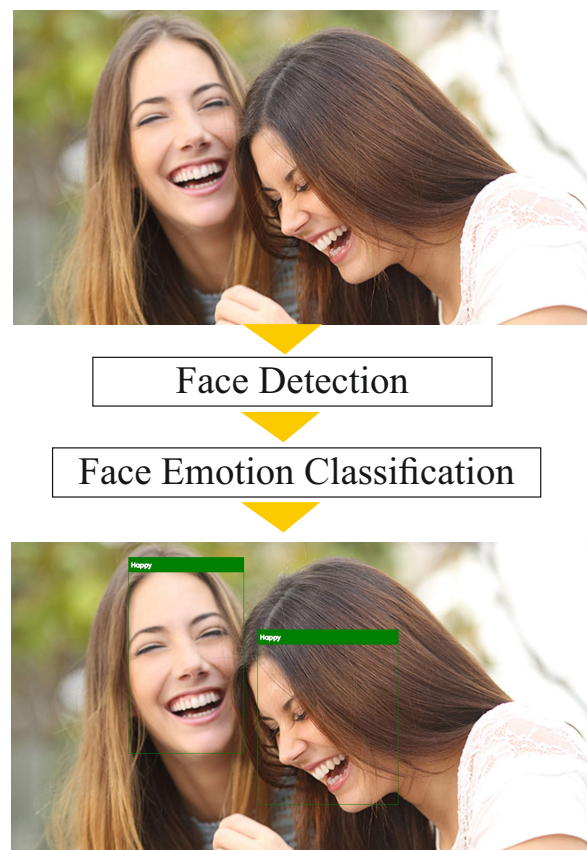


FIGURE 1.2: Major components in the two-stage real-time face emotion recognition.

The work in this study develops a high-performance face emotion recognizer that is powerful and efficiently operates within reasonable processing time. A deep learning network is utilized for both architectures to acquire high precision and

overcome several challenges. However, the general problem of this method is the computational overhead cost. Efficient architecture overcomes this issue while compressing excessive learnable parameters. Moreover, the specific architecture with the combination of attention increases the accuracy of the prediction without adding significant parameters. On the other hand, the performance of CNN models also tends to be influenced by other factors such as data augmentation, training strategy, and loss function. This work also uses several tweak techniques to improve the network's performance.

1.3 Contributions

The work in this research explores various approaches to developing a complete real-time facial emotion recognition framework. It proposes a face detector and facial emotion classification on low-cost computing devices to build a high-efficient system. The first part of this study introduces real-time face detection to find the facial location. The work contributions are as follows:

- A novel high-performance face detector uses lightweight CNN architecture to fast localize faces region in real-time and is implemented on low-cost devices.
- A novel face detection architecture employs a backbone to discriminate facial elements, and a multi-level prediction supports this detector to estimate the multiple face locations on different scales.
- The proposed face detector was conducted comprehensive evaluations on benchmarks, showing that the model reaches comparative performance against the state-of-the-art CPU detectors.
- This efficient face detection is fastest than other low-cost detectors, real-time operating on the low-cost devices.

The second study offers an efficient facial expression classification to predict the emotion of the face region of interest. The study contributions are as follows:

- A novel light backbone architecture presents efficient feature extraction using a partial transfer approach that rapidly filters specific elements with fewer parameters than the baseline model.

- A cascade attention network selects the interest features in a series configuration at the end of the backbone. The module sequentially enhances the valuable features to increase the accuracy of the facial emotion classifier without producing high computation costs.
- The proposed model achieves competitive accuracy against the state-of-the-art model on benchmark datasets. Additionally, this model can smoothly operate at a processing data speed of 90 FPS on a CPU device.
- The integration of face detection and facial expression network can effectively recognize human emotions emphasizing effectiveness and efficiency. integrated model can run in real-time at 45 FPS on a CPU device.

1.4 Disposition

This part describes the organization of this manuscript. Section 2 discusses various methods related to deep learning models to detect and classify facial expressions. It contains the classical and the current techniques that present face detection on the CPU using single and hybrid approaches. It also discusses several influential works on facial expression.

Section 3 discusses the proposed architecture of a low-cost face detection network. It includes the general architecture, proposed backbone, detection module, anchor method, and training strategy. The last part of this section contains the evaluation of the proposed face detector compared to other CPU models.

Section 4 explains the proposed architecture for face emotion classification. This section contains several parts related to the proposed model, including an efficient backbone, sequential attention, connection, and classifier module. Furthermore, it also examines the proposed model on several datasets.

Section 5 discusses the proposed model integration for high-efficiency facial expression recognizers on low-cost devices. It includes the hardware setup and discussion on the speed test and visualization results in real-case scenarios.

Finally, Section 6 concerns the possible future research direction and conclusion of this research work.

Chapter 2

Literature Review

The scope of this thesis considers three central components, including face detection, facial expression, and attention mechanism. This section discusses several publications on these issues that influence the recent studies.

This section includes three parts. The first part concerns several methods affecting the current CPU-based face detection. Then, the second part presents several works related to facial expression recognition. Finally, the last one describes attention mechanism methods in a deep learning model.

2.1 Vision System

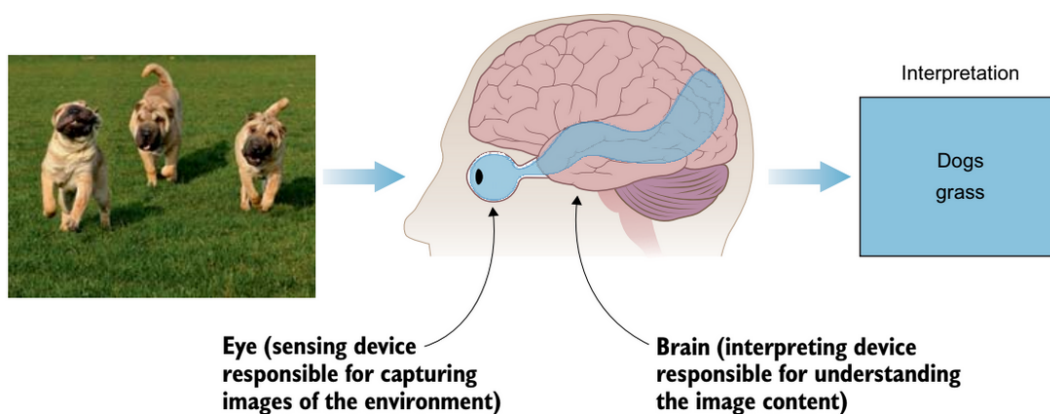


FIGURE 2.1: The process flow of human vision (Elgendy, 2020).

Visual perception is a process of scanning patterns from an object that tries to create a system. It can help to understand the environment based on visual input (Elgendy, 2020). Traditional image processing techniques are not entirely accurate. Machines can process images with different understandings, and it is not a trivial

task. The visual system works like humans, animals, and most living organisms see things. It utilizes sensors or eyes to capture images. It then uses the brain to process that information and interpret it, as illustrated in Figure 2.1. The system then issues an image prediction based on the extracted data.

Computer vision is inspired by the human vision system and has been able to copy visual abilities to machines in recent years. Figure 2.2 shows that computer vision also requires two main components: a sensory device like the eye and a robust algorithm as a brain to interpret and classify image content.

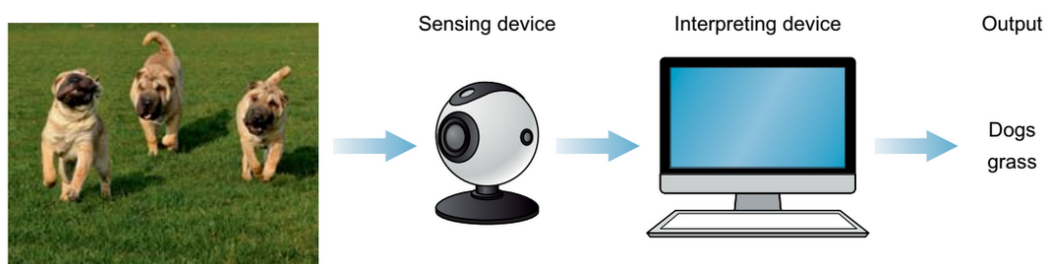


FIGURE 2.2: The parts of computer vision utilize sensing and an interpreting device (Elgendy, 2020).

2.2 Image Classification

Image classification is a computer vision task that categorizes groups of pixels in an image. It predicts the image's content label by applying a particular method. Categorization rules can use one or more features and texture characteristics. Therefore, it predicts class content based on feature information extracted in an image. It requires feature extraction to determine feature characteristics and obtain specific information as important data for the classifier to decide the categories.

The learning-based classification system shows a more satisfactory performance than the conventional approach. The learning-based classification approach consists of supervised using labeled datasets and unsupervised learning analyzing and classifying unlabeled datasets.

2.3 Object Detection

Object detection is used to localize objects through the input image. This research has been growing in the past few years. Machine learning contributes a lot to improve its performance. Object detection generally consists of feature extraction and classification tasks (Kurnianggoro, 2019). This localization task utilizes bounding boxes as the output of the detection results for each object.

Figure 2.3 illustrates the basic principle of an object detection method. It consists of two-stage methods. The feature extractor is applied initially to find useful features from an area in the image. Region candidate of the object provides its location information. Then, a classifier predicts the region, whether it is a specific object or not, based on the input features. The detector repeats all processes from the top-left until the bottom right frame. A sliding window can adjust its scale, making it possible for the detector to predict objects in different scales.

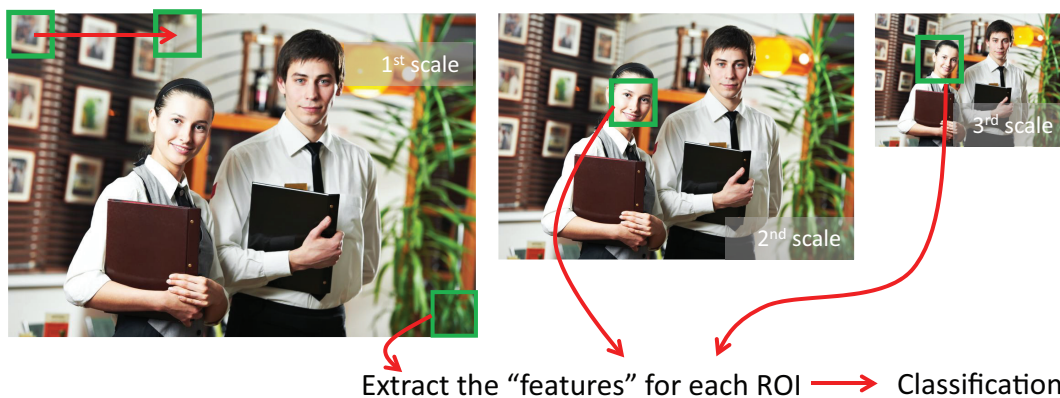


FIGURE 2.3: An object detector with the sliding window strategy (Kurnianggoro, 2019).

2.4 Classical Face Detector

A conventional method applies feature engineering and machine learning to extract information from pixels and selects important features of an image. Both tasks are crucial elements in classical object detection. Feature engineering plays the leading role in discriminating against distinctive features, while machine learning helps detectors amplify important features through a continuous learning process. A detector is generally designed from a combination of feature design and classifier selection.

Viola&Jones approach (Viola and Jones, 2001) has succeeded in robustly localizing faces and claims that it produces less processing time. This detector can operate at a real-time speed that is applied on low-cost devices. Haar-like features extracted important features while the AdaBoost classifier predicted those features as facial features. The role of the Integral image is to compress processing time so that the detector can work quickly. An integral image computes the total sum over a given region by accessing four locations in the summed area table. This operation results in a more efficient computation than standard calculations of haar features.

Haar-like features compute the difference between light and dark regions compared to one or more regions nearby. It utilizes pre-defined templates containing rectangular combinations. Furthermore, the integral method is applied to each region to calculate the total score. This technique shortens the computation time, allowing the Viola&Jones face detector to operate efficiently.

On the other hand, AdaBoost machine learning is used to select a particular Haar feature and adjust its threshold value. This approach combines weak classifiers to generate strong classifiers. This combination is an efficient series filter for interpreting the characteristics of an object. Classifiers are arranged into a cascade. The greatest weight filter is inserted at the beginning of the stage to remove non-face images quickly. If an image area fails to pass one filter during the classification process, then that area is immediately classified as non-face. Thus, the predictable area of the face must successfully pass through the entire filter process.

Another work of face detector has utilized template matching and skin-color information to localize the frontal face (Jin et al., 2007). It segmented the eye area instead of applying segmentation to all image regions containing information from the three-dimensional position, orientation, and lighting conditions. Then extract the skin-color information and apply normalization to the candidate's facial areas. Template matching helps the detector to classify facial areas with normalized data distribution.

Remarkable classical face detection has also been presented by (Ban et al., 2014), which offers a skin-color probability approach through a boosting algorithm. It serves to emphasize skin color features and ignore non-skin color knowledge. The color distribution is implemented in YCbCr space to separate the luminance and chrominance. The histogram displays the difference in the intensity of information

contained in an image area. It supports the performance of the Bayes rule to establish the probability of a color vector. Candidates of the facial area are trained by the skin boosted cascade to separate and select the valuable skin color distribution.

A semi-local structure pattern (SLSP) has been proposed as a feature extractor approach based on a set of binary patterns at local region-based differences (Jeong, Choi, and Jang, 2015). It solves the problem of illumination variations, distortion, and sparse noise. The SLSP can encode the comparison of the center features with local neighbors of the surrounding area. Furthermore, the AdaBoost feature selection method trains specific features extracted in a region to predict facial regions.

The complicated background in an image causes a conventional face detector to produce a high false positive. Therefore, (Kang, Choi, and Jo, 2016) has offered a skin color modeling to enhance the region-based classifier performance that separates facial and background features. In addition, a sliding window efficiency helps the whole algorithm to be able to reduce the processing time so that it can operate quickly. It reduced time to a maximum of 47% from the standard sliding window method. The combination of skin modeling and segmentation strengthens the color characteristic selector and emphasizes noise reduction to discriminate against interest features.

2.5 Deep learning-based Face Detector

The face detectors using Convolutional Neural Network (CNN) are presented to overcome challenges that are difficult to handle by the classic method, including challenges in scale, position, distortion, and background. A single-shot scale-invariant face detector (S3FD) uses CNN architecture that develops the SSD (Single-shot detector) model for localizing faces with small object variations, huge amounts of negative anchors, and few amounts of anchors that match with the face. In addition, the S3FD also introduces several approaches to alleviate these problems. It applies six face detection layers of varying sizes. The Conv3_3 layer is used to predict small faces with a feature map size of 160×160 , while the other detection layers work on the Conv4_3, Conv5_3, and three more layers at the extra convolution layers at a stride of 32, 64, and 128. Figure 2.4 illustrates the architecture of S3FD.

The S3FD uses varying anchor sizes that match the size of the receptive field. It

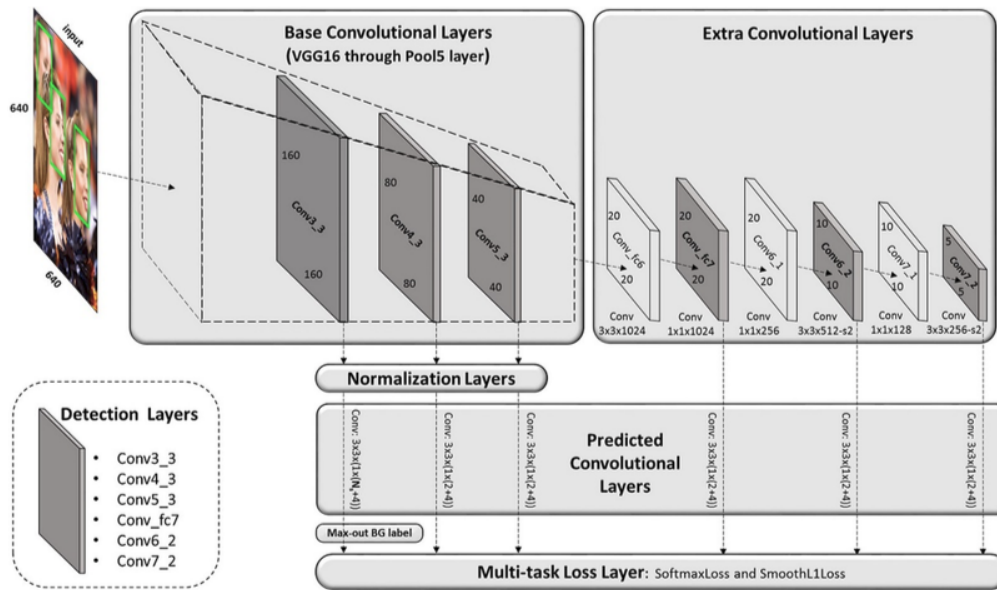


FIGURE 2.4: A S3FD architecture (Zhang et al., 2017).

effectively uses an anchor size of 16, 32, 64, 128, 256, and 128 on prediction layers. The first detection layer contains about 75% anchors, but the background contains more than 99.8%. This problem creates an imbalance between positive and negative classes in training. In order to overcome this issue, S3FD discriminates against the background with the maximum value. It assigns each prediction layer to predict one foreground and n -background. Then it selects the maximum value as the final predictive score and sets another score to be foreground.

The dual-shot face detector (DSFD) is a face detection architecture that applies an FPN-like model to improve the relationship between features of different frequencies (Li et al., 2019). The feature pyramid network (FPN) is implemented to improve the prediction of the object on multiple scales (Lin et al., 2017). It is proposed to fuse low-level with high-level features to enrich the object information extracted by the convolution layer. In addition, it also implements an enhancement module (FEM) to improve the quality of the feature combination. It uses upsampling and convolution techniques to equalize the different channel sizes at each level. A single convolution layer adjusts the channel size, and a dot product operation is used to aggregate two maps with distinct levels. Furthermore, the dilated convolution was applied to three branches with different amounts stages and combined all outputs with concatenating technique. DSFD applies predictions on basic and enhanced features. So this strategy gets better results when it is compared to the other single-shot detection

architectures.

A TinaFace detector (Zhu et al., 2020) proposes a robust architecture with a simple baseline to localize various human faces, as shown in Figure 2.5. It uses ResNet-50 (He et al., 2016) to extract facial features and discriminate against trivial features. In order to improve multi-scale detection performance, TinaFace applies a six-level Feature Pyramid Network to filter out the features on different scales. It employs an Inception module to enrich the receptive area inserted in each detection head. Regression and classification heads apply five layers of FCN series to generate final predictions. This model achieves high accuracy on the WIDER dataset, which is highly efficient and effective in overcoming these challenges.

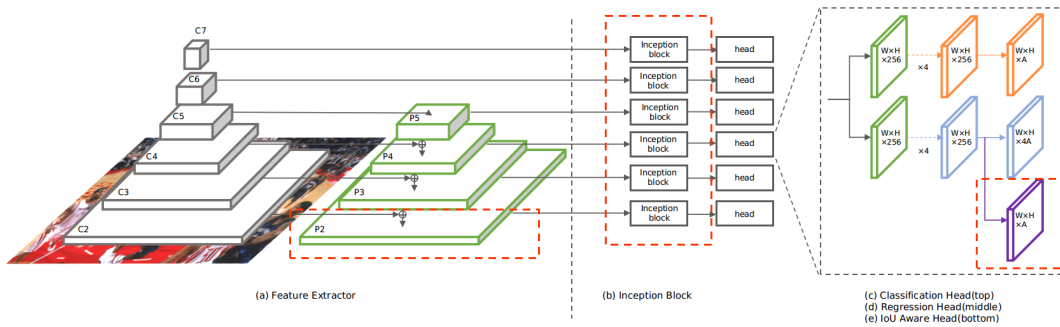


FIGURE 2.5: A TinaFace architecture (Zhu et al., 2020).

Recently, YOLOV5 was introduced as an object detection tool that can work effectively and efficiently. YOLO5Face (Qi et al., 2021) made a few modifications to YOLOv5 and implemented it as a face detector. It applies CSPNet (Cross Stage Partial Network) (Wang et al., 2020) to extract essential facial features efficiently, as shown in Figure 2.6. In addition, YOLO5Face uses Spatial Pyramid Pooling to increase the receptive area and split the essential elements. The FPN structure is also employed to aggregate features with different frequencies. A feature aggregate network is applied to each neck to enrich information by fusing FPN features with the backbone. YOLO5Face provides a variety of architectural sizes with different computations, parameters, and performance scores that are competitive with other competitors. It is claimed to be a capability of the detector capacity that can adapt to hardware implementation.

A benchmark is utilized to quantify the performance of face detection methods. Wider face provides various challenges and is categorized into three face sizes: large, medium, and small. There are more than 50 methods are competing with each other

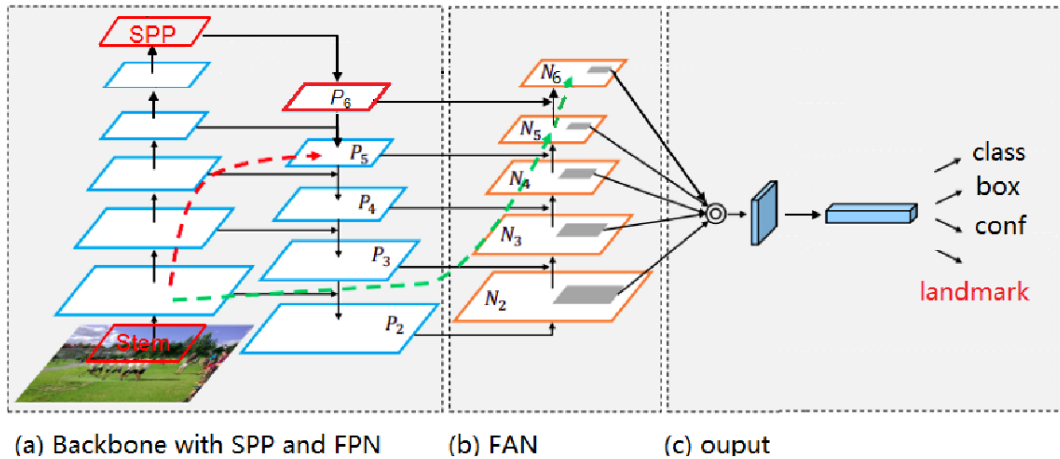


FIGURE 2.6: A YOLOV5Face architecture (Qi et al., 2021).

including traditional models (Yang et al., 2014; Mathias et al., 2014; Viola and Jones, 2001) and deep learning models (Chi et al., 2019; Najibi et al., 2017; Zhu et al., 2018; Cai et al., 2016; Yashunin, Baydasov, and Vlasov, 2020; Liu et al., 2021). The evaluation shows that the recent deep learning models outperform the traditional models.

2.6 Facial Expression Network

Facial expression classification is a method for predicting human facial gestures that represent human emotions. It is influenced by interconnected facial components and composes a specific texture to describe the relationship between these features. Previous work has applied a conventional approach to recognizing human expressions. However, this method achieves low accuracy even for non-frontal face challenges. Traditional feature extractors are not robust in discriminating against similar facial features to the background and feature occlusions.

Deep learning-based methods show high performance as feature extractors to discriminate specific facial components. (Otberdout et al., 2020) has proposed deep covariance descriptors to identify human expressions. The input face encodes using local and global covariance descriptors that occupy at the symmetric positive definite (SPD) manifold, as shown in Figure 2.7.

This approach uses the Gaussian kernel and Support Vector Machine (SVM) to generate a proper positive-definite on the specific manifold and classify static facial

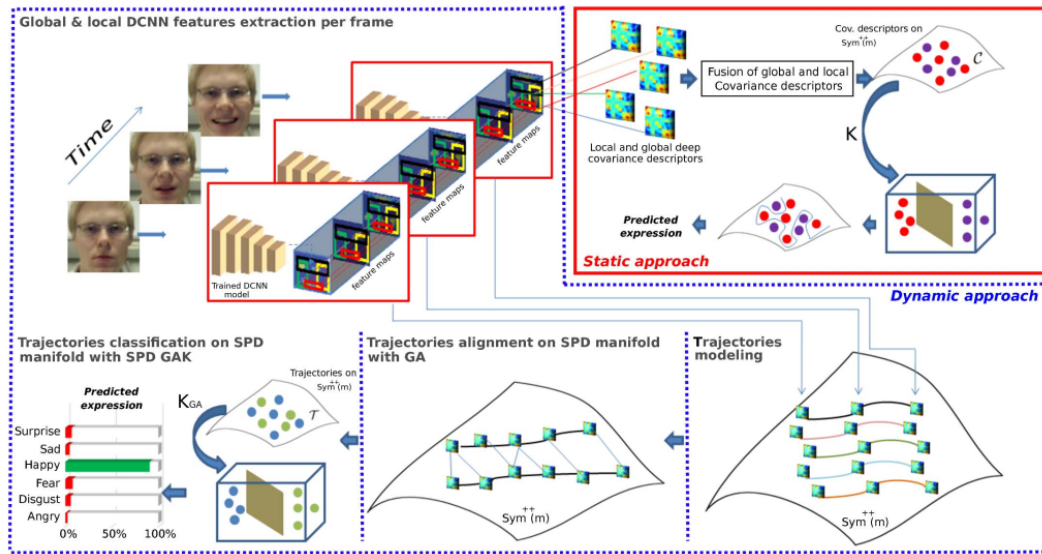


FIGURE 2.7: The ExpNet approach (Otberdout et al., 2020).

expressions, respectively. In order to align trajectories, it is helped by Global Alignment (GA) using the Log-Euclidean Riemannian Metric (LERM). The experimental results show that high performance is achieved by this deep approach on three public datasets.

Other work has obtained high accuracy by offering an ensemble architecture with multi-level convolutional neural networks that predict seven emotion classes (Nguyen et al., 2019). The robust feature extractor was developed from VGGNet by utilizing 3×3 filters to optimize performance and efficiency. This architecture ignoring the fully connected and setting the channel size on the entire feature map is a power of two for computational reasons.

The multi-level network aggregates the final feature map at the third to fifth stages to obtain a fusion of middle and high-level features. It emphasizes varied information rather than using only high-level frequencies for classification. This ensemble model achieves increased accuracy by combining three parallel networks and applying a different combined feature variation to each network, as shown in Figure 2.8. However, this architecture produces a lot of parameters and an expensive computational cost.

MLCNN offers a 3DCNN configuration on video input for extracting features through image sequences. It applies the ensemble of MLCNNs as the backbone to distinguish facial features for all faces from the background. A temporal model

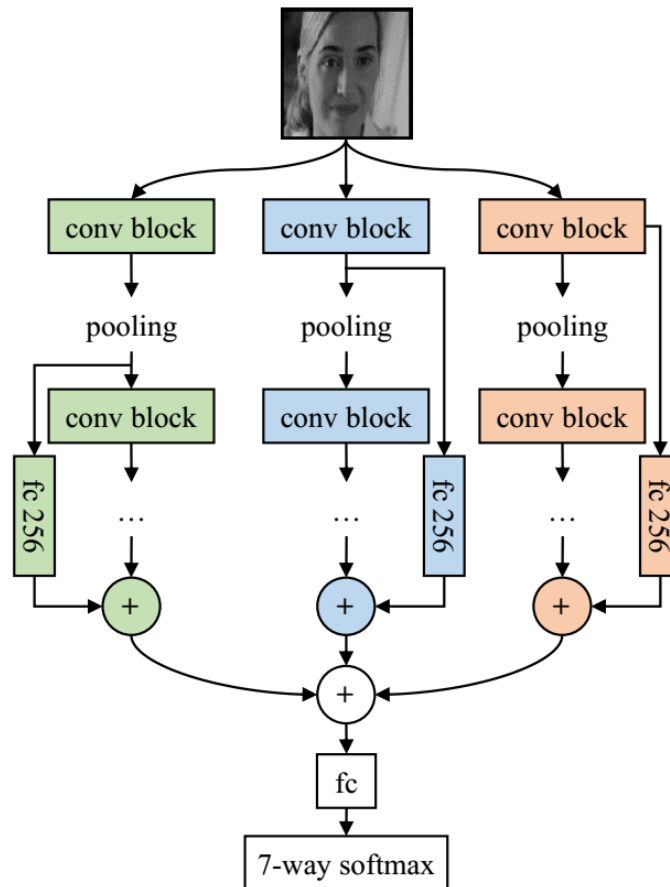


FIGURE 2.8: Overview of the ensemble of MLCNNs approach (Nguyen et al., 2019).

utilizes convolutional and pooling to extract the information and applies softmax activation to generate the prediction.

Efficient architecture has been introduced to classify facial expressions using Hierarchical Deep Neural Network Structure (Kim et al., 2019). Feature extraction highlights the appearance of facial components combined with geometric features. LBP (Local Binary Pattern) computes the divergence of interest features with the background through the difference between center features and local neighbors. Then the geometric feature-based model learned to extract the action units (AUs) information based on coordinates. It recognizes the muscle movement characteristics and relates them to facial expressions predictions.

In order to improve its performance, it integrates the probabilities result of the two features with considering the second-highest prediction error. Additionally, this work generates neutral emotion by applying an autoencoder architecture to extract dynamic features between neutral and peak expression.

2.7 Attention Mechanism

A lightweight model generates a few parameters and a low computational cost. It generally employs a small number of kernels and channels. Nevertheless, this architecture tends to achieve low accuracy. A lightweight model is not robust to discriminate valuable features. Therefore, it requires additional modules, including the attention block (Fu et al., 2019). The attentive module distinguishes specific features and reduces unimportant elements (Cao et al., 2019). It can also highlight several interest features and interprets the weighted score. It is applied to improve the important information from the input features (Hu, Shen, and Sun, 2018).

The attention network is already presented by previous work to improve the performance of the backbone in facial expression recognition. An attention visual-based approach was utilized after the VGG-16 network to increase the accuracy score (Sun, Zhao, and Jin, 2018). It offered an attentive CNN architecture consisting of three main modules: local convolution, region of interest, and aggregation of local features with a prediction layer.

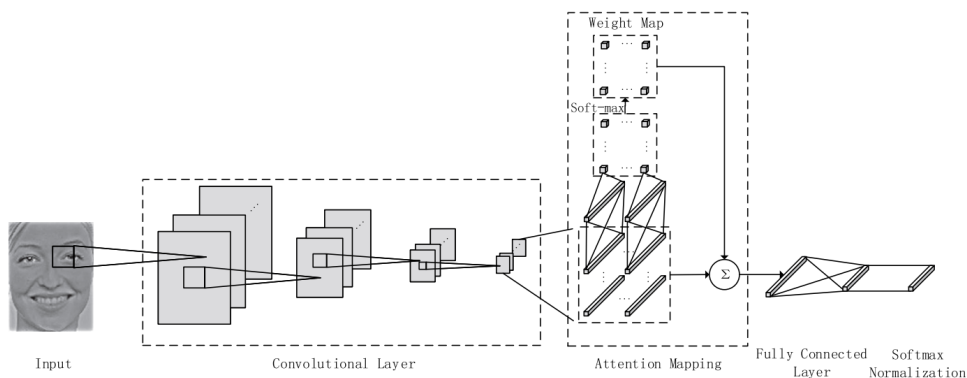


FIGURE 2.9: The CNN with visual attention (Sun, Zhao, and Jin, 2018).

An eleven-layered CNN with Visual Attention was employed to extract specific facial features. This baseline is inspired by the VGG network and improves its accuracy with visual attention at the last stage, as illustrated in Figure 2.9. It adds an enhancement block before the dense module. It is assumed to improve high-level features quality by applying weighted probability. The softmax function helps this module by normalizing feature information so that each vector denotes the feature at a specific location.

A work utilized a Multiple Attention network (MA) to filter specific facial components for face expression tasks (Gan et al., 2020). It simulates humans' coarse-to-fine visuals to increase convolutional performance as extractor features. The attention mechanism can learn discriminative features. A region-aware sub-net finds locally important areas that describe features of interest. Meanwhile, the ERSnet module comprehensively discriminates against these features by applying multiple attention.

The MA block plays a crucial role in aggregating various references using the learned masks. It employs a hybrid block on a branching sub-module, containing learned region attention. Besides, each mask is learned comprehensively to capture expression characteristics. Furthermore, it is diffused with a feature map extracted by the backbone, namely a weight learning branch. These sub-branches adaptively extract the critical regions that provide intensive global attention to the features of interest.

(Li et al., 2020) employed an attention module with a superficial configuration on a combined extraction. This architecture contains four main components: the feature extractor, the attention block, the reconstruction block, and the classifier module. It combines RGB and LBP inputs to increase the variety of feature textures. VGG-16 Net discriminates against both input maps, which have a strong transfer learning ability. A 13-layers of this baseline extracts deep features and then reduces the dimensionality to the same size of both inputs.

An attention mechanism is applied after the twin backbone module to find valuable features that help increase prediction performance, as illustrated in Figure 2.10. Different regions are assessed by the attention score that represents the most useful features. The trunk and mask branches highlight specific features by comprehensive learning of different input textures. Then, it applies an element-wise product to aggregate the two sections and generates refined feature maps. In order to adjust the attention map, it implements a reconstruction module that utilizes dense connection convolution layers. An atrous convolution is used to capture large receptive fields without increasing the number of kernel parameters. Besides, a double backbone and a reconstruction module add extra computational cost.

An architecture of deep CNN relies on a graphics processor to operate fast at the inference phase. Besides, an attention module can increase the performance of the

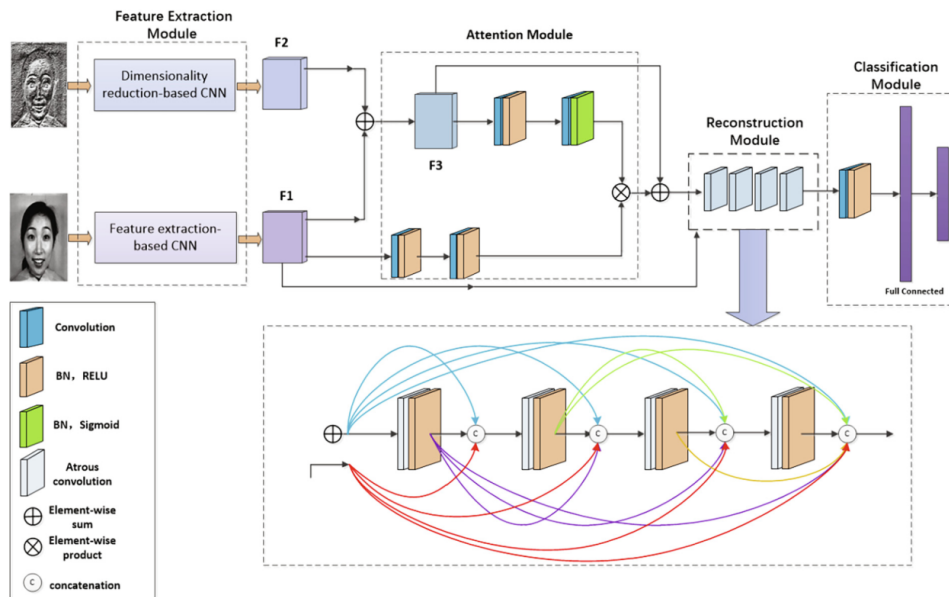


FIGURE 2.10: The architecture of attention mechanism-based CNN network (Li et al., 2020).

extractor feature. Therefore, the proposed model offers a robust sequential attention network to enhance the lightweight extractor ability. It also retains the real-time speed of the system to increase its efficiency.

Chapter 3

Face Detection Network

Face detection is a basic computer vision approach to predict the face's location through an image. This method screens the facial area against the background, enabling the facial emotion recognizer to improve the accuracy in a real-world scenario. Conventional methods have been introduced with problems of inaccuracy in difficult challenges (Viola and Jones, 2001, Jin et al., 2007, Ban et al., 2014, Jeong, Choi, and Jang, 2015 and Kang, Choi, and Jo, 2016). On the other hand, deep learning as a modern method can significantly increase accuracy. Convolutional neural network extracts feature objects robustly by distinguishing important facial components from background features. Several methods adopted the most of the recent object detector architecture (Zhang et al., 2017 and Li et al., 2019). They apply FPN to aggregate features at different frequency levels that apply multiple prediction layers to assign varying anchor dimensions.

The deep CNN architecture emphasizes the precision of essential feature prediction by employing a large number of convolution filters (Zhu et al., 2020, and Qi et al., 2021). Lightweight parameters and cheap computation do not support the high performance of the method. They generate a lot of learnable parameters. In addition, They also used many operations to employ neuron nodes that inflict computational overhead. These weaknesses require that these detectors have a dependence on accelerator devices, so they tend to work slowly on cheap devices such as CPUs and with the heavy cost of processing time. Implementing a deep learning method in real-world scenarios is the biggest issue for application developers.

FaceBoxes (Zhang et al., 2019) and Densely Connected Face Proposal Network (DCFPN) (Zhang et al., 2018) accurately localize faces on a CPU running at real-time

speed. FaceBoxes employs a light architecture to decrease the sizes of maps and extract the information using Rapidly Digested Convolution Layers. It sets a series of convolution and pooling with large strides. C.ReLU was employed to activate neuronal output by reducing the number of output channels. This method is claimed to improve computational efficiency significantly. Multiple Scale Convolution Layers (MSCL) apply an inception block to enrich the variety of information from the feature map. In addition, this module plays a role in predicting multiple scales of faces. The features pyramid technique is involved in the architecture to merge the feature information of objects of various sizes. FaceBoxes achieves excellent precision and is able to operate at 28 FPS on CPU devices.

DCFPN has offered a CNN architecture for face detection with high accuracy and CPU real-time speed. It uses a robust convolution network and an anchor matching strategy to increase the precision rate of small objects. In addition, it explores the performance of a proper L1 loss function to evaluate the predicted boxes that localize small faces. DCFPN can detect multiple faces at 30 FPS on a low-cost device in 640×480 resolution.

Both detectors can only run smoothly on CPU devices with high clock rate specifications. So the efficiency of these detectors still depends on relatively expensive devices. The work of this manuscript is proposed an efficient and accurate face detection that can work on low-cost computing devices with CNN-based light architecture. The architecture includes superficial layers, which use sub-blocks and specific configurations to avoid reduced performance. This work focuses on two things. Firstly, it designs a CNN-based architecture by applying convolution layers and an efficient structure without compromising detection performance. Secondly, the training strategy improves detector performance without increasing consumption and operation time, such as the balancing loss function, augmented image techniques, and parameter configurations.

3.1 System Architecture

The proposed detector is designed with two essential modules, including a backbone and a prediction layer. It leverages the robustness of CNN as a feature extractor by applying a small number of the filter compared to regular architectures. The

feature extractor using CNN architecture has delivered excellent performance by producing high precision and accuracy. However, this requires high computation to acquire a slow data processing speed. In order to avoid this problem, the proposed extractor utilizes a superficial layer that filters the spatial information maps. Therefore, this model emphasizes improving real-time speed and maintaining detection performance.

3.1.1 Backbone Module

The backbone module plays an essential role in sequentially extracting features that take advantage of feature learning. The main block can discriminate against important facial components from the background. This module implements shrink block rapidly declines spatial maps with maintaining essential elements. It shrinks the feature map dimension with local extracting at the different stages. On the other hand, this block focus prevents the increase in overhead computation and the heavy-weight of kernel usage on large feature maps. The four convolution series is applied to capture specific features while keeping the quality of feature information. Fig. 3.1 shows that this module reduces a map dimension to 32 times less than a source size.

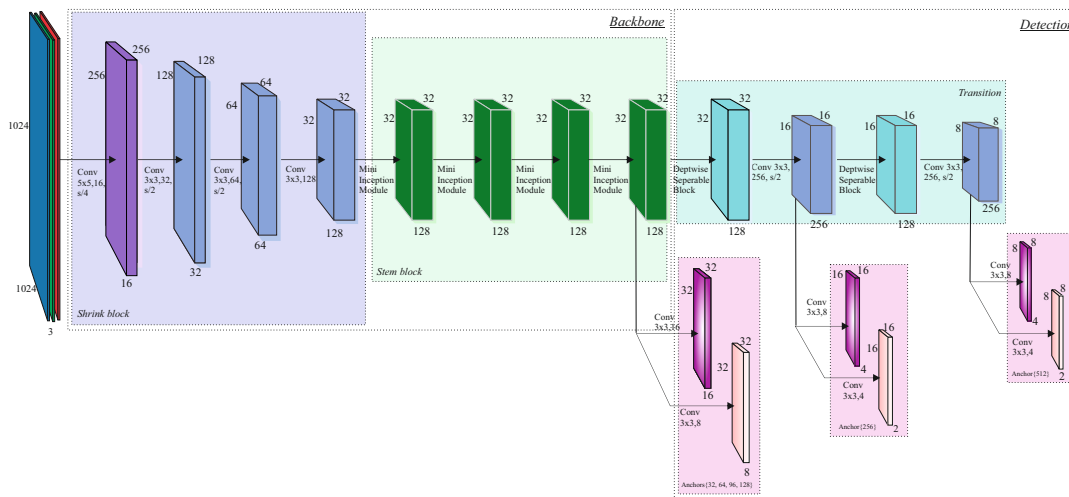


FIGURE 3.1: Architecture of the face detector network.

In order to fast reduce the maps, it applies a large filter size at the beginning of the process and then uses the 3×3 kernel in the remaining phase. This approach can save memory usage and increase detector speed by applying a big kernel at the initial stage. Instead of using the same stride size, it sets 4, 2, 2, and 2 to quickly and drastically reduce the map. Furthermore, the training time is accelerated using batch

normalization, which helps this detector obtains optimal performance. In addition, rectified linear units (ReLU) are also applied in the neurons by eliminating negative scores linearly.

The proposed detector applies stem block as an important sub-module to fully discriminate facial features. It sequentially employs four mini-inception that is more efficient than the common version. This module extracts object features by increasing the variety of the receptive field (Szegedy et al., 2015). Moreover, it generates cheap parameters to catch information on different receptive areas. Several works implemented this block in the real-time application for object detection and image recognition task (Lee et al., 2017; Jiang et al., 2019).

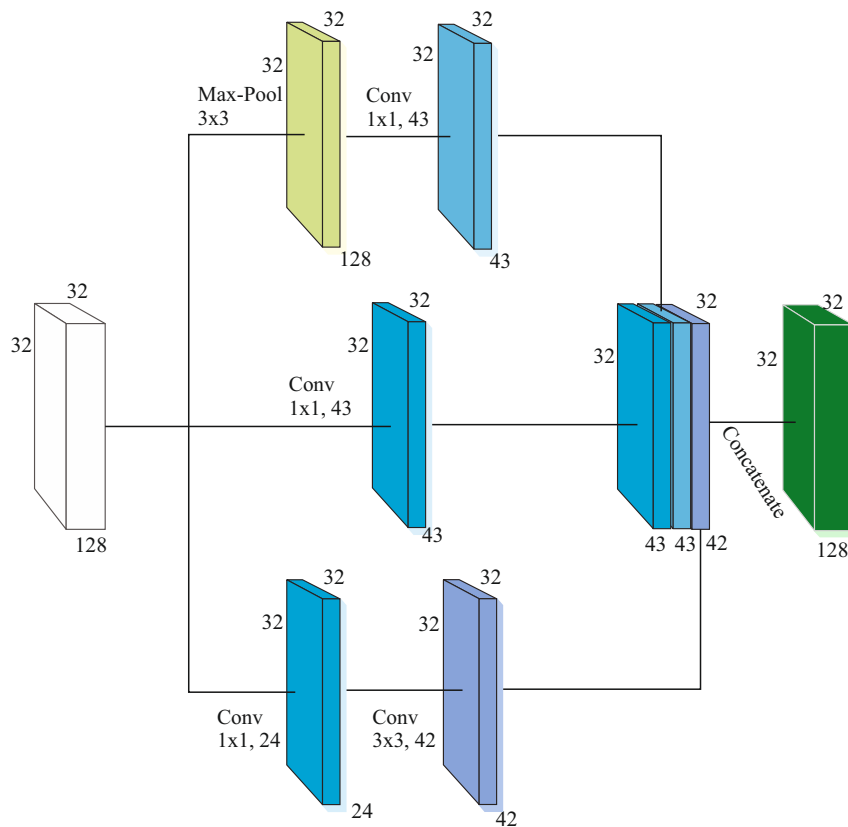


FIGURE 3.2: Mini-inception module for face detector network.

Instead of applying extensive filters or a lot of branches, the proposed module avoids high computation by producing an informative map. It takes advantage of simple element characteristics that comprehensively help a shallow network select feature interests. To improve performance, it implements more than one block, which simultaneously applies the convolution and pooling processes, as shown in Fig. 3.2. It combines the feature map developed by a combination of convolution and

pooling to produce valuable information. This aggregation enriches the knowledge of the coverage area of different filters, which increases face detection accuracy.

3.1.2 Detection Module

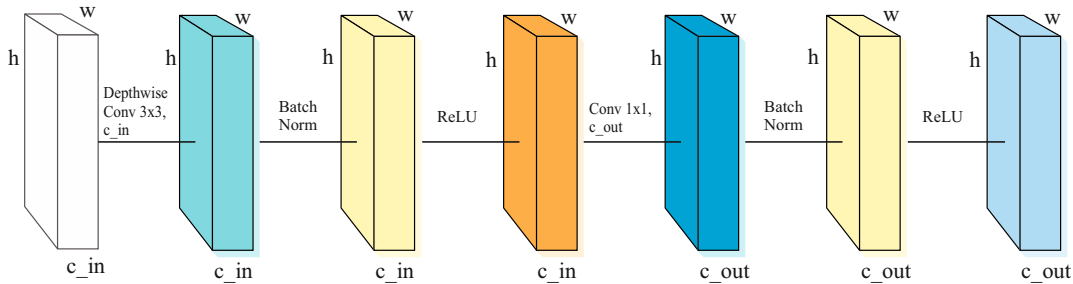


FIGURE 3.3: Transition block for face detector network.

Several approaches have been introduced to estimate region proposals prospects in various object scales. In order to produce an abundant number of parameters, the detector implements a pyramidal feature hierarchy that involves multi-layers prediction and clusters the face candidate based on the size.

Transition block. This detector uses a transition block to transform feature map sizes between multiple prediction layers. In order to save computation, it applies simple blocks to keep the information of the features map. It avoids applying pooling to decline the map dimension. Fig. 3.3 shows that it bridges important features maps at different levels by convolution with stride two. It is more potent than a pooling operation. Hence, it adopts the depthwise separable convolution (Howard et al., 2017) to extract high-level features, and a 3×3 convolution is used to reduce the map dimension. The convolutional block is more robust than a scalar filter. It can be assumed that the transition block sets the specific map scale by increasing the amount of information. This module accommodates three predictors with different sizes, transferred to a head network to assign the prediction result.

Multi-level detection. Object detector commonly utilizes the detection layer to estimate the location of objects at the last framework. This high-performance detector applies multi-level detection to predict different face scales, addressing the limitation of a single predictor. The specific receptive area is unable to accommodate variations in the size of the face. Besides, information about the small object is over-reduced in the last network, resulting in the loss of valuable features.

The proposed detector uses the pyramidal feature hierarchy to avoid extra computation from the Feature Pyramid Network (FPN). It establishes different feature map sizes to accommodate different object scales. Additionally, it employs a variety of scaled anchors that will adjust the size of the predicted box. In order to support multi-scale prediction, the module employs three levels that assign it according to face dimension. It allocates small, medium, and large faces at 32, 16, and 8 prediction map dimensions, respectively. The diversity of anchor sizes can accommodate the variation of the face scale in the specific prediction layer.

3.1.3 Anchor Strategy

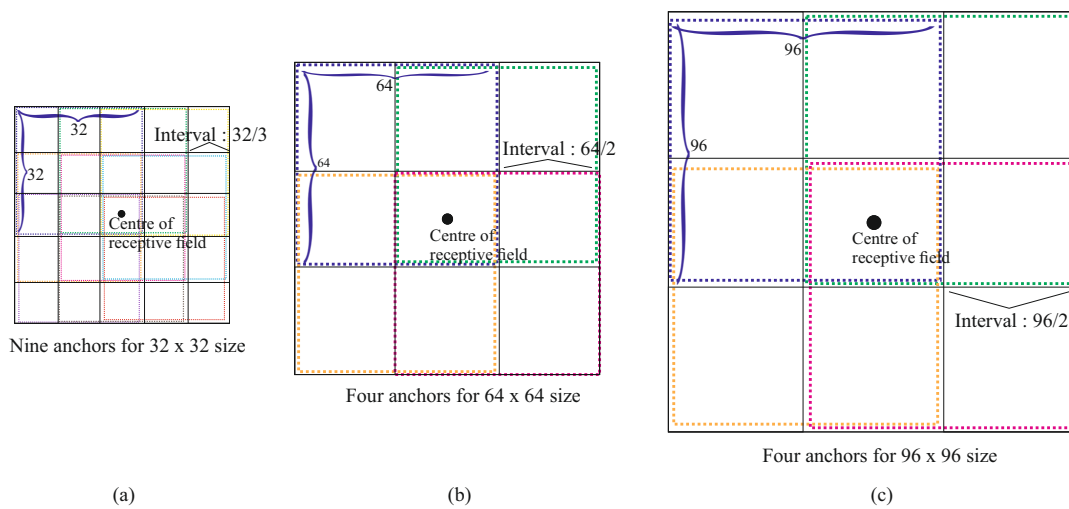


FIGURE 3.4: A approach of anchor density with 32×32 (a), 64×64 (b), and 96×96 (c) dimension.

Initial scale information is defined using the anchor technique in estimating the size of the predicted bounding boxes. It is employed to increase the prediction precision based on dimensional clusters. In order to capture represented multiple-scale faces, it allocates not one type of anchor box. According to this assignment, multi-square anchors are assigned to various detection modules. The proposed detector utilizes anchor densification to boost the precision of small predicted objects (Zhang et al., 2018). It involves the same scale neighboring anchors with specific intervals. It uses center point information to define the distance between boxes. However, it adds a new scale of 96 to reduce the gap on small faces, which contrasts with the original version.

Fig. 3.4 illustrates that this strategy is applied only to small anchors to avoid redundant computation. This approach employs 9, 4, and 4 anchors with 32, 64, and 96 scales at the first detection block, respectively. In addition, single anchors with 128, 256, and 512 are applied to all prediction layers.

3.1.4 Balanced Loss Functions

Generally, an anchor-based object detector generates regression and class scores. It specifies the coordinate parameters, size boxes, and class of each predicted anchor. On the other hand, a learning method requires measuring predictive that quantifies the difference between the actual value with the reference. It helps to boost the performance of weight neurons by minimizing errors. The proposed detector applies multi-boxes loss that combines the regression and classification error by assigning it to each anchor. However, the imbalanced score has a problem that causes only one function to work optimally. Therefore, it offers a balancing function to address the issue by associating the error prediction. The loss function is expressed as:

$$L(cls_i^n, reg_i^n) = \frac{\delta}{Pos} \cdot \sum_i L_{class}(cls_i^{pred}, cls_i^{gt}) + \frac{\eta}{Pos} \cdot \sum_i L_{reg}(reg_i^{pred}, reg_i^{gt}), \quad (3.1)$$

where reg_i^{pred} is prediction vectors that indicate as coordinates and size of box for each i -th anchor and class n , reg_i^{gt} is the reference box from the dataset, cls_i^{pred} is the predicted label class, and cls_i^{gt} is the reference label class from annotation dataset. The Pos is the number of matched boxes with ground truth, and it is used δ and η , as the balancing constant of 2 and 1, respectively. $L_{class}(cls_i^{pred}, cls_i^{gt})$ is a classification error using softmax-loss, illustrated as:

$$L_{class}(cls_i^{pred}, cls_i^{gt}) = - \sum_{i \in Pos} m_i^{pred} \text{Log}(cls_i^{pred}) - \sum_{i \in Neg} \text{Log}(cls_i^0), \quad (3.2)$$

where m_i^{pred} is a matching logic of each initial box to reference that defines as 1 or 0, and cls_i^0 is the non-object probabilities. Additionally, this detector applies Huber loss in the regression part:

$$L_{reg}(reg_i^{pred}, reg_i^{gt}) = \sum_{i \in \{x, y, w, h\}} Hr(reg_i^{pred} - reg_i^{gt}), \quad (3.3)$$

in which

$$Hr(x) = \begin{cases} 0.5x^2 & : |x| < 1 \\ |x| - 0.5 & : otherwise \end{cases} \quad (3.4)$$

is a smooth error to calculate the difference of both scores optimally. The equation describes that input less than 1 will be affected by a soft effect using the quadratic function.

3.2 Training Dataset and Implementation Details

In the training phase, the model uses the WIDER FACE dataset to obtain feature knowledge. This benchmark provides 32,203 images that include the training image of 12,800. The variation of data is used as learnable information of facial elements. In order to increase the various instance, it applied the augmentation technique. It is involved in reducing the overfitting problem. The augmentation includes sequential operations of the random crop, scaling, horizontal flipping, and color distortion. The training input size is generated by resizing the cropped patch to a scale of 1024×1024 .

In its implementation, the proposed network involves several optimization settings simulated in the PyTorch framework. It defines random weights in the initial phase through end-to-end training. Then, each neuron's weight will be updated using Stochastic Gradient Descent (SGD). It sets a momentum of 0.9 and weight decay of $5 \cdot 10^{-4}$. We use various learning rates at three epoch stages to optimize the training process. It uses 200 epochs with a 10^{-3} learning rate at the first stage, 50 epochs with a 10^{-4} learning rate at the second stage, and applied 10^{-5} learning rate with 50 epochs at the last phase.

The training process employs 32 batch sizes to split the entire image dataset into small groups. Moreover, it also requires a matching process that determines the IoU (Intersection over Union) of predicted boxes and ground truth. This parameter defines 0.5 to select a set of anchors predicted to be the best box.

3.3 Experiments and Results

The performance of the face detector is explained in this section by investigating each module in model analysis and evaluating the detector on the AFW (Zhu and Ramanan, 2012), PASCAL face (Everingham et al., 2010), FDDB (Jain and Learned-Miller, 2010), and WIDER FACE (Yang et al., 2016) dataset.

TABLE 3.1: Model analysis of each module.

Modules	Proposed detector						
Balanced loss	✓						
Anchor densification	✓	✓					
Transition	✓	✓	✓				
Multi-level prediction	✓	✓	✓	✓			
Shrink module	✓	✓	✓	✓	✓		
Stem module	✓	✓	✓	✓	✓	✓	
TPR on FDDB (%)	97.00	96.70	96.40	96.30	90.40	89.50	90.10
Inference time (ms)	18.87	18.86	18.20	18.10	17.62	16.93	20.42
Parameters (K)	989	989	892	888	915	827	896
Computation (MFLOPS)	195	195	166	166	158	90	111

3.3.1 Model Analysis

The ablative experiments are comprehensively conducted by describing each performance and efficiency to quantify the ability of each block. It replaces the proposed block with a standard module and evaluates the true positive rate (TPR). We also measure the inference speed of each configuration model using Core i5 with 640×480 resolution. This experiment utilizes the same training configuration to obtain a fair comparison. The accuracy uses a true positive rate with 1,000 false positives on the FDDB dataset, as shown in Table 3.1. Firstly, the experiment is examined that removes the balanced loss by setting a constant of 1 for δ and η , respectively. This experiment shows that this loss can improve TPR by 0.3% and not influence inference time, parameters, and computational complexity.

Furthermore, it examines the anchor strategy’s impact by replacing it with the default type. It means that only uses one anchor on each scale. This experiment decreased TPR by 0.3% and inference time of 0.66 ms. Thirdly, the transition block is replaced with 1×1 convolution. This investigation only declines 0.1% TPR but speeds up 0.1 ms. A single detection module applies to the detector that replaces multi-level detection. It only uses a last prediction layer and stacks all anchors in

this layer. The multi-level detection module can significantly increase TPR by 5.9%, reducing the time by 0.48 ms. Besides, this replacement causes a declining usage of the floating-point operations and increases the number of parameters.

Fifthly, an experiment replaces the convolution layer with max-pooling on shrink blocks. It confirms that the convolution operation effectively improves the TPR by 0.9%. However, it adds the time consumption by 0.69 ms. Additionally, the convolutional block also significantly increases the number of parameters and operations. The last experiment substitutes a mini-inception in the backbone with common inception. The mini-inception drops the true positive rate by 0.6% while reducing the time by 3.5 ms. The proposed inception block uses a smaller number of kernels with a parallel configuration that is more efficient than the original.

3.3.2 Comparison with other detectors

The proposed detector is evaluated on several benchmark datasets, such as AFW, PASCAL face, FDDB, and WIDER FACE datasets, which compare the performance with other competitors.

AFW dataset. This dataset contains various Flickr images containing 203 pictures with 473 labeled faces. It provides face challenges, including position, accessories, and expression. In addition, it also covers different lighting and background variations. The proposed detector performance is compared with other commercial and research works. Fig. 3.6 illustrates that our architecture leads the competitors. It is better than 0.28 of FaceBoxes. Qualitative results show that multiple faces with diverse challenges can be accurately located in the area, as shown in Fig. 3.5 (a).

PASCAL face dataset. This dataset is created by selecting from the PASCAL VOC dataset, including 851 pictures with 1,335 annotated faces. It provides a multi-pose face challenge with various environments and backgrounds. Fig. 3.7 illustrates that the presented detector is leading to competitors. The accuracy is 1.02 higher than the latest CPU detector (FaceBoxes). Fig. 3.5 (b) also presents this detector can detect face in different illumination scenarios.

FDDB dataset. This dataset provides various faces from famous people that contain 2,845 pictures, including 5,171 labeled faces. Entire images are obtained from Yahoo websites covering multi-pose, illuminance, scale, and background challenges.



(a)



(b)



(c)



(d)

FIGURE 3.5: Visualization of the detection results on AFW (a), PASCAL face (b), Fddb (c), and WIDER FACE (d).

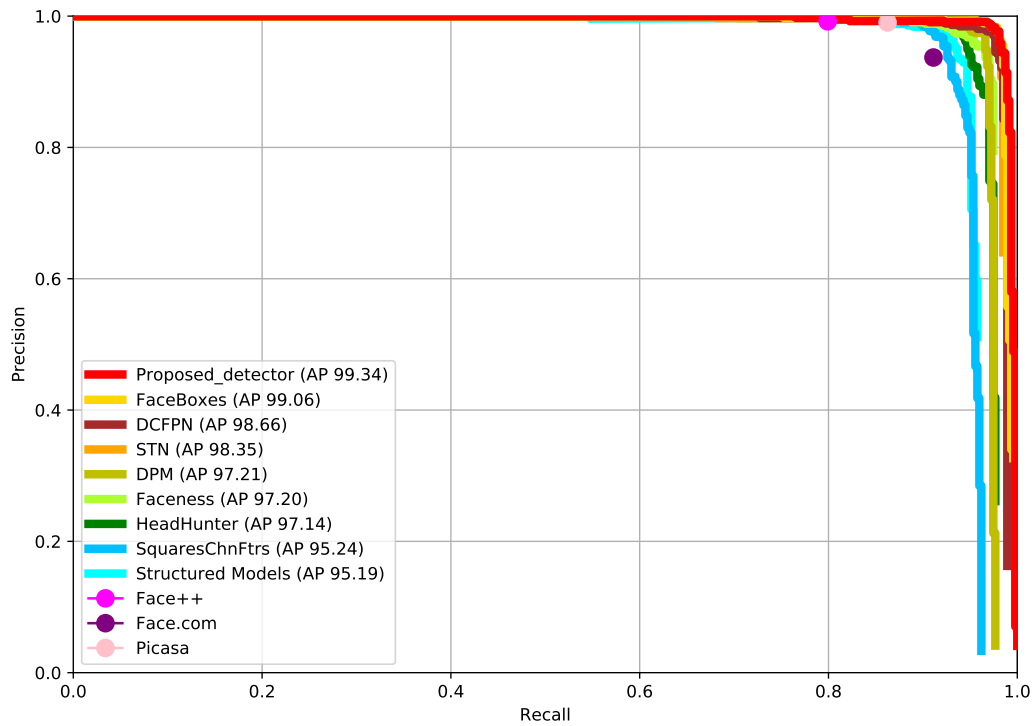


FIGURE 3.6: Comparison of Average Precision (AP) on the AFW dataset.

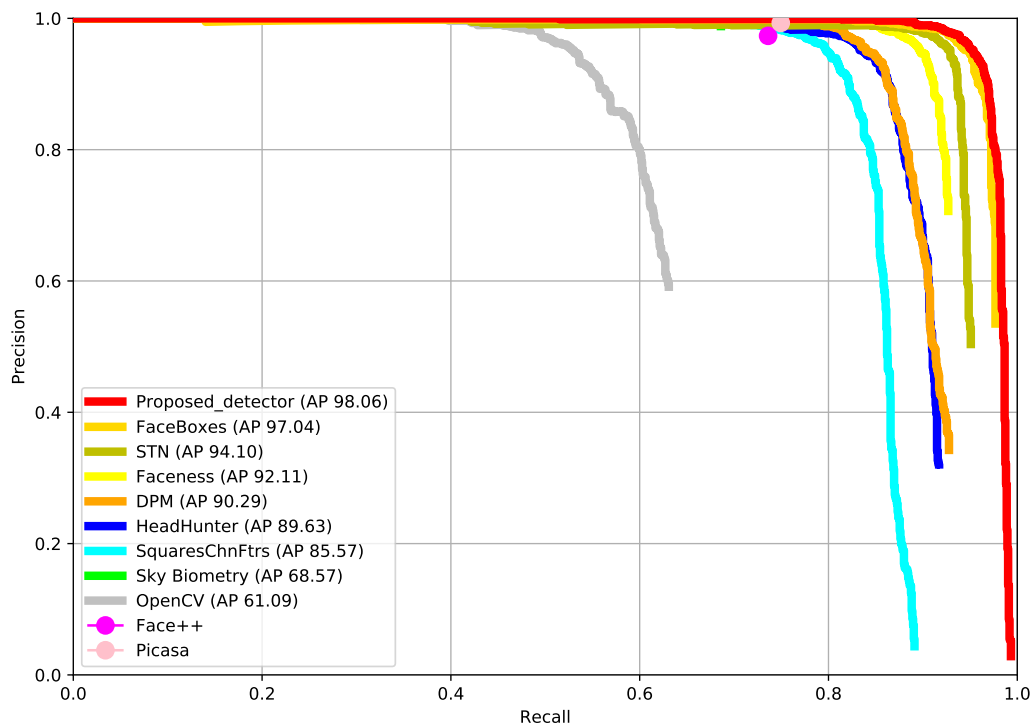


FIGURE 3.7: Comparison of Average Precision (AP) on the PASCAL face dataset.

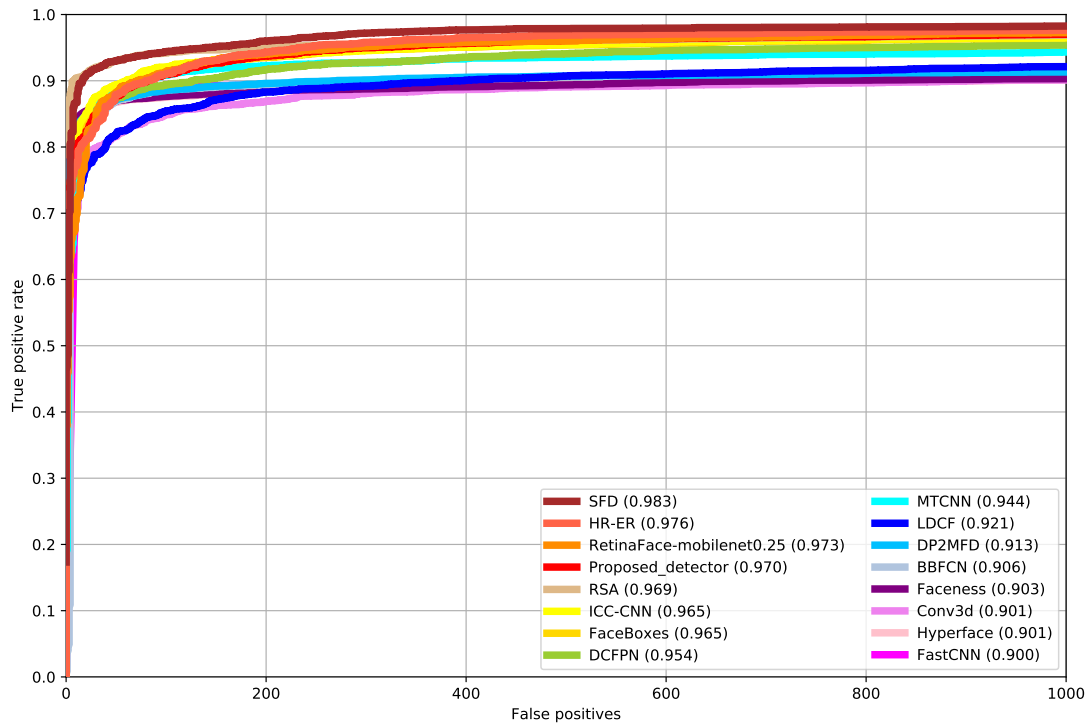


FIGURE 3.8: Comparison of true positive rate uses discrete ROC (Receiver Operating Characteristics) curves on the FDDB dataset.

As shown in Fig. 3.5 (c), the proposed detector can localize faces in the occlusions challenge. In this case, the evaluation uses discrete prediction, comparing the IoU of the predicted box and the ground truth. It establishes the score is one when the IoU ratio is higher than 0.5 and 0 otherwise. Fig. 3.8 illustrates that the detector achieves competitive performance with the RetinaFace-mobile version. It only differs 0.3% below this competitor. However, the proposed detector achieves better accuracy than FaceBoxes and DCFPN. The proposed detector achieves lower average precision than HE-ER (Hu and Ramanan, 2017) and SFD (Zhang et al., 2017), but these detectors are the heavy model that produces an expensive computation cost. So it can be concluded that they are not feasible for real-time performance on a CPU.

WIDER FACE dataset. This dataset provides several difficult challenges that are collected from real-world scenarios. It covers unconstrained human faces with varied scales, multiple views, occlusions, expressions, and different illumination. Generally, it is separated officially into 40% for training, 10% for validation and 50% and testing sets. The evaluation sets provide three difficulty levels for a fair evaluation: easy, medium, and hard. Tiny faces with occlusion are the most difficult challenge of this subset of datasets. The visualization in Fig. 3.5 (d) shows that the proposed detector can accurately localize the multi-facial region, even for partially covered

faces.

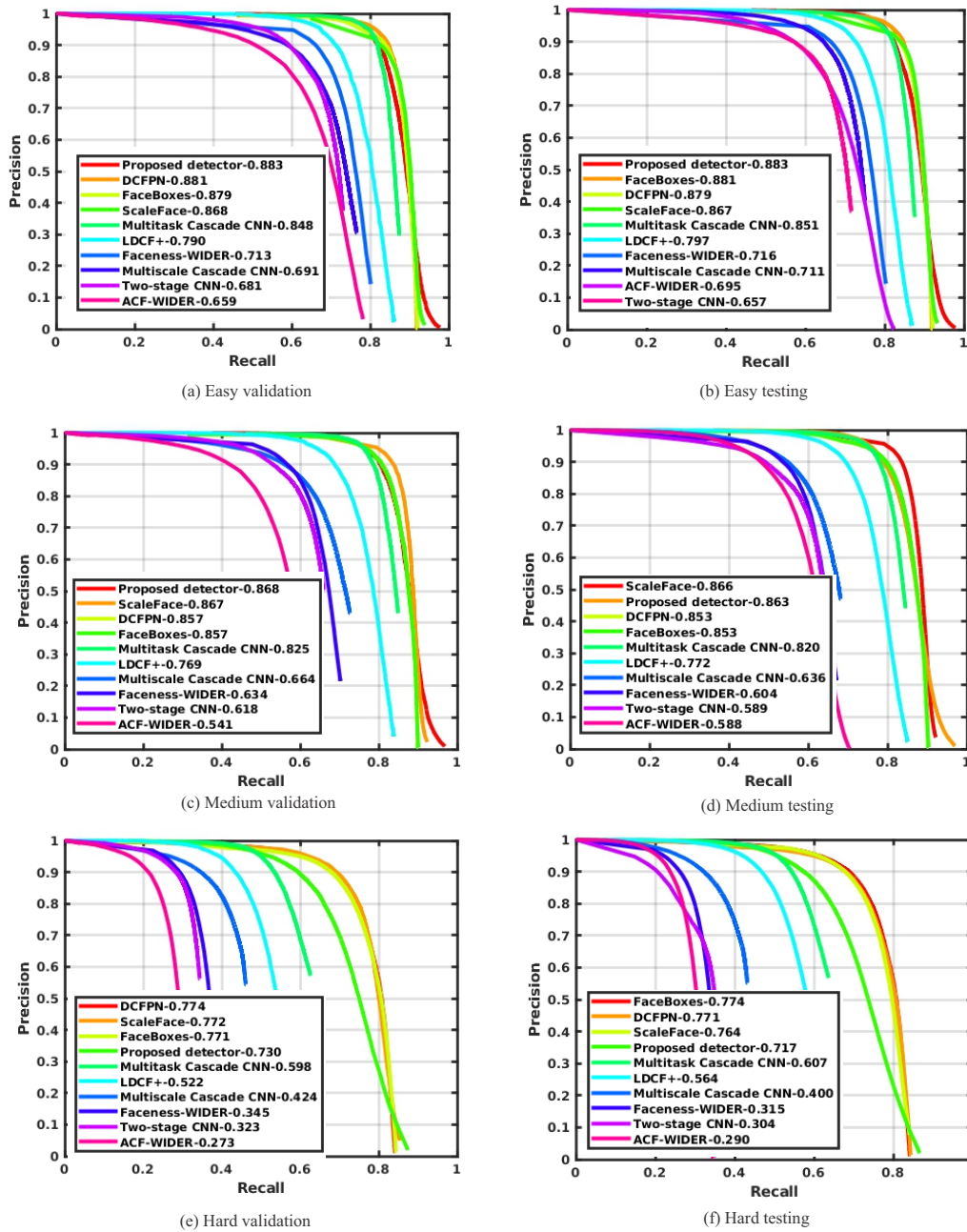


FIGURE 3.9: Comparison of Average Precision (AP) on the WIDER FACE dataset.

As shown in Fig. 3.9, the proposed detector achieves average precision of 0.883, 0.868, and 0.730 for easy, medium, and difficult validation sets, respectively. This model also reaches 0.883 for easy, 0.863 for medium, and 0.717 for hard testing sets. The proposed detector achieved more excellent performance than FaceBoxes on the easy and medium criteria. However, this competitor is superior to the hard category. The proposed wider-0 method is not robust for identifying tiny faces in the low-layer

features. A shallow network employs feature extraction that weakly discriminates small essential elements. Additionally, ScaleFace (Yang et al., 2017) obtains more accurate results on the medium and hard testing criteria. However, this model slowly operates in real-time, requiring the accelerator GPU inference stage.

Chapter 4

Facial Expression Network

In this section, the architecture of facial expression classification is explained in detail by comprehensively describing each proposed module. It utilizes a convolutional neural network model, which can strongly discriminate the vital elements against trivial features. Deep learning networks are the most popular approach to recognizing human facial expressions in an input image. It can robustly distinguish the specific facial features given the meaningful special gesture of each predicted expression. This architecture tends to employ deep feature maps by applying convolution operations with a large number of the filter. It causes heavy computational and parameter impact. Therefore, Deep Convolutional Neural Networks operate with large time inference and depend on expensive devices. The proposed network implements a light backbone and attentive module to efficiently extract the interest facial feature so that it supports an emotion recognizer to operate at real-time speed on an inexpensive device. This section explains an efficient network for fast facial emotion classification. As shown in Fig. 4.1, the general framework contains an efficient extractor, an attention module, and a predictor.

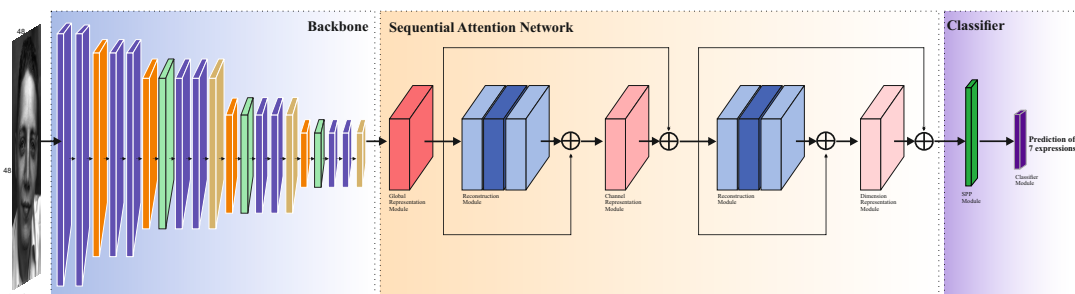


FIGURE 4.1: The general architecture of facial expression network.

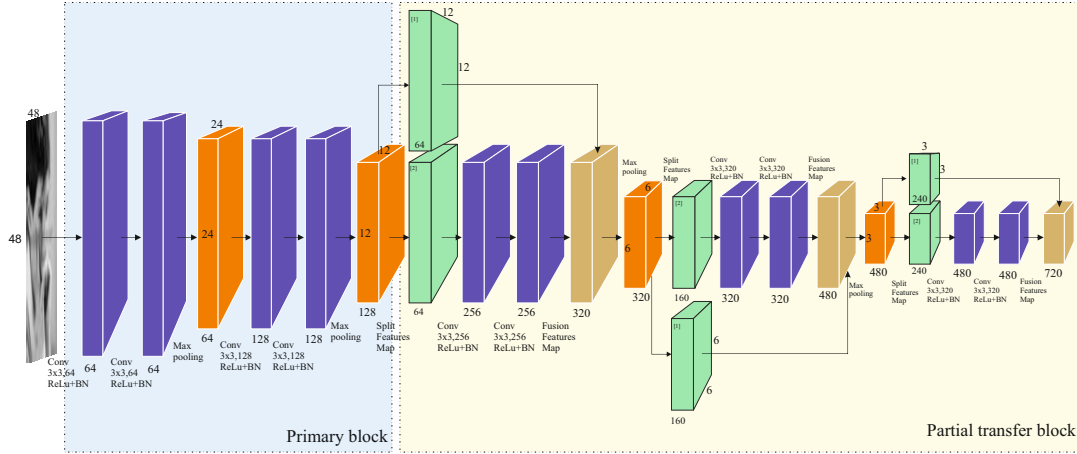


FIGURE 4.2: Architecture of efficient backbone with transferred partial feature.

4.1 Backbone module

A CNN-based architecture employs the updated filter to distinguish essential features as a powerful feature extractor. The use of deep layers obtains high accuracy of feature selection, but it requires expensive computational costs. It even produces a large number of parameters. So this architecture tends to operate slowly on low-cost hardware. Therefore, this work presents an efficient backbone to help the system achieve fast real-time speed. VGG-13 architecture (Simonyan and Zisserman, 2014) was developed by applying 3×3 convolution as the optimal filter. This architecture includes five stages and applies batch normalization and rectified linear units (ReLU) after a convolutional block. It uses five max-poolings to decline the feature map dimension. Furthermore, the transferred partial approach is applied to VGG-13 to save the parameter and computation.

The proposed backbone offers the Efficient Partial Transfer (EPT) module to rapidly discriminate between interesting and trivial facial components. Fig. 4.2 shows that this module includes two blocks, such as a primary and a partial transfer block. The 3×3 kernels are employed sequentially on the primary block, and maximum pooling is employed in every stage to decrease the map size efficiently. The transfer module divides a input map x_i into two chunks $[x_0^s, x_1^s]$. Then, a sequential convolution extract features from a first segment, while another chunk is transferred to the last convolutional block. The EPT backbone fuses (\oplus) the two feature maps enriching the different information frequencies, as illustrated as:

$$EPT = C_2(C_1(x_1^s)) \oplus x_0^s. \quad (4.1)$$

Two convolutional layers ($C_1(\cdot)$ and $C_2(\cdot)$) with ReLU and batch normalization are utilized as the extractor block for core segment(x_1^s). The EPT block is inserted in the third to fifth stages to reduce computations from using a large number of kernels. It also focuses on getting the frequency level variations on mid and high-level features that provide more complex information than low-level features. Additionally, this efficient backbone module produces fewer parameters than the VGG-13 architecture. The EPT module emphasizes using an extractor in the partially input feature map with less computational complexity than the standard block. However, it maintains the precision of the model by combining the information from multi-level frequencies.

4.1.1 Sequential Attention Module

The attention module enhances the medium and high-level features caused by a shallow backbone that is not powerful to distinguish these crucial elements. It offers a cascade structure that can select more comprehensive specific features to improve map quality. This architecture delivers a more satisfying accuracy than the parallel one, as shown in Table IV. The attentive block can capture interest elements and selectively boost the intensity. It presents three sequential blocks to highlight global and local representation. Besides, it also uses a transition module to bridge each attention module.

Global attention module. A feature of an object tends to have a strong relationship with other elements on a map. Convolution operation has a limited view, which only extracts spatial-based features. Therefore, the global attention module helps the network catch the global context of specific information, combining it with selective excitation features. It powerfully filters out long dependencies to enhance the relationship between facial features and expression. This module combines global context (G_x) (Cao et al., 2019) and excitation modules (E_x), that can be expressed as

$$GA_i = Gx_i + Ex_i. \quad (4.2)$$

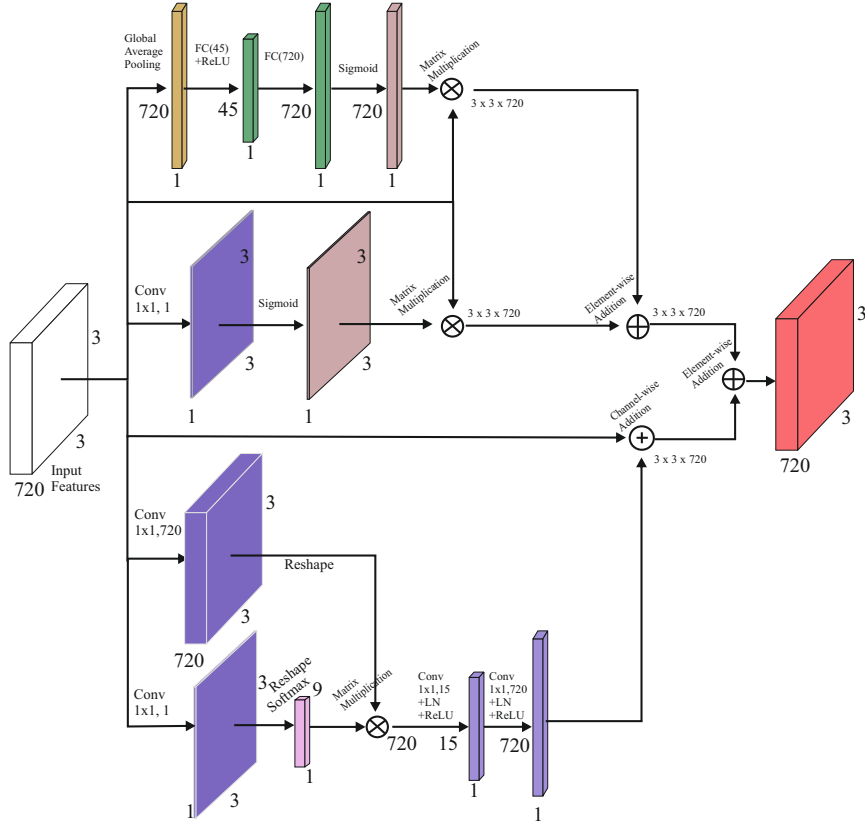


FIGURE 4.3: Global attention module.

Each representation module operates at the i^{th} query position and combines each element of the module output at the same position and coordinates. The global representation module finds contextual information (Tx_i) from the input features (x_i) with $H \times W \times C$ dimension, generating the weighted score the be illustrated as

$$Tx_i = \sum_{i=1}^{HW} (x_r \cdot \frac{\exp^{W_j x_i}}{\sum_{n=1}^{HW} \exp^{W_j x_n}}), \quad (4.3)$$

where x_r is reshaped convolutional map ($W_i x_i$) with $HW \times C$ dimension. In order to reconstruct the extracted element, a global attentive module also uses a bottleneck block that involves Γ as the ReLU function and \mathbb{N} as layer normalization. Finally, this module fuses the representation map to the input features with addition operation, formulated as follows:

$$Gx_i = x_i \oplus \Gamma(\mathbb{N}(W_{u2}\Gamma(\mathbb{N}(W_{u1}Tx_i)))). \quad (4.4)$$

The proposed attention module employs an excitation block that merges the selected feature of a simple attention and feature representation based on pooling. It

utilizes a Global average pooling (GP) to extract the statistical information, while a convolution generates simple attention. The excitation block is described as

$$Ex_i = x_i\sigma(W_i x_i) + x_i\sigma(W_{u2}\Gamma(W_{u1}GP(x_i))), \quad (4.5)$$

where it generates the probability weights in the two branch module using sigmoid (σ). The fusion module increases the intensity of specific features by selectively capturing essential contexts. Additionally, this combination module strengthens the global representation of features related to each facial emotion.

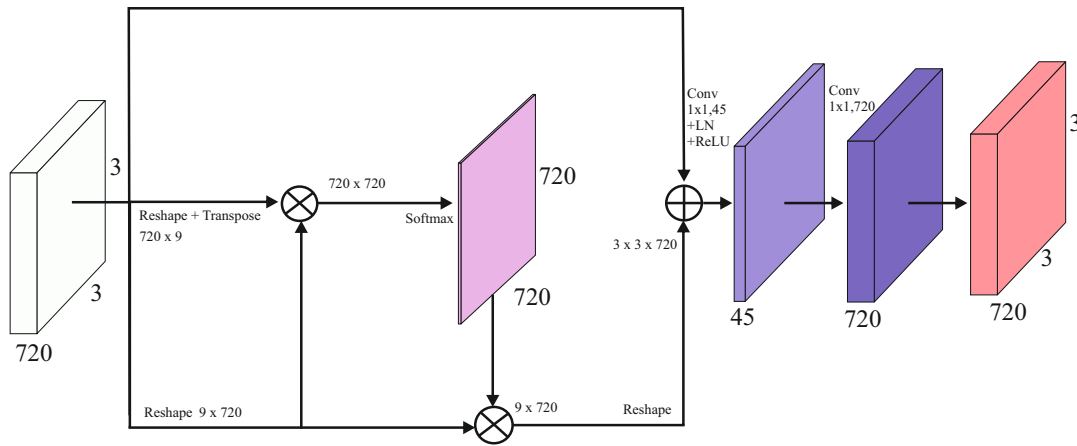


FIGURE 4.4: Channel attention module.

Channel attention module. This block captures the representative facial features according to the input channel size, highlighting channel dependencies by producing valuable weights. It maps a set of essential features related to channel size by assigning a high score to the valuable pixels. Those weights are applied to input features to update unclear features. It adopts the work of (Fu et al., 2019) and improves it with the bottleneck module at the end of attention, as shown in Fig. 4.4. This module multiplies the transposed (t_j) with reshaped tensor (t_i) to generate the feature mapping ($C \times C$), which can be illustrated as

$$h_i = x_i + \sum_{i=1}^C \left(t_i \cdot \frac{\exp(t_j \cdot t_i)}{\sum_{i=1}^C \exp(t_j \cdot t_i)} \right). \quad (4.6)$$

It normalizes the weights on the channel map by using Softmax activation, which is used to obtain the representative tensor. Then, channel-based attention is aggregated with the sum of all weights with the input map (x_i) to update the represented

features. The proposed module employs a bottleneck technique to extract and reconstruct the facial features that can be expressed as

$$CA_i = W_{v2}\Gamma(\mathbf{N}(W_{v1}h_i)). \quad (4.7)$$

The approach contains a convolutional series and applies a small filter in the first layer. It aims to reduce the number of parameters by using fewer filters. In addition, this module also plays a role in selecting the represented features. The channel representation module combines attention and reconstruction blocks to discriminate against specific facial features. It can increase the intensity of the relationships between elements through channel-based mapping. Each channel provides various information of intensity. This diversity can be used as the knowledge of the model to learn the expression characteristic.

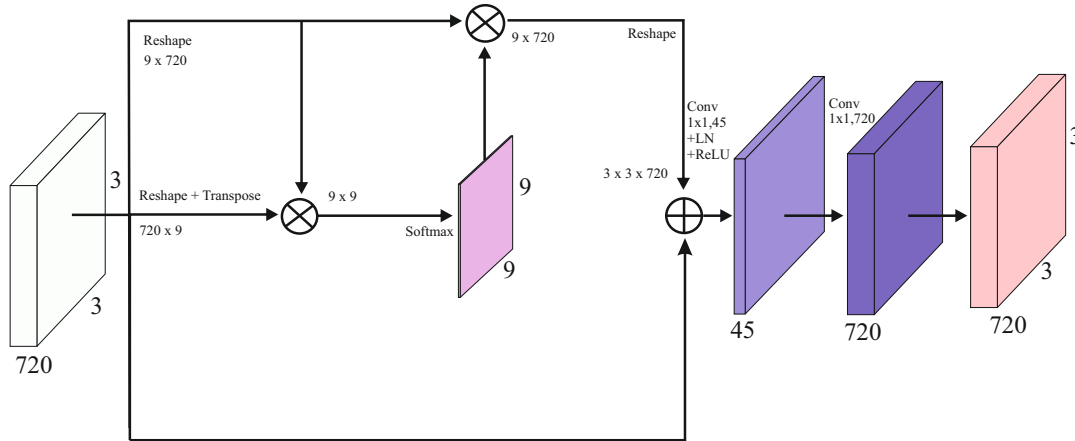


FIGURE 4.5: Dimension attention module.

Dimension attention module. The proposed module improves discrimination performance to be more selective in estimating local facial features related to their expressions. As shown in Fig. 4.5, it raises the ability to project a long-range contextual into a positional mapping. In order to reconstruct the represented features, this implements a bottleneck block at the end of the attention module. This block is described as

$$DA_i = W_{v2}\Gamma(\mathbf{N}(W_{v1}p_i)), \quad (4.8)$$

where p_i is a valuable feature represented from the dimension-wise attention, it generates a relation map with dimensions of $HW \times HW$ by multiplying between the

reshaped (t_i) and transposed tensor (t_j). It describes the relationship between features by finding the similarity score between pixel positions according to the input dimensions. The attention module generates a weighted score by applying the Softmax normalization. The dimension attention module can be defined as follows:

$$p_i = x_i + \sum_{i=1}^{HW} \left(\frac{\exp(t_i \cdot t_j)}{\sum_{i=1}^{HW} \exp(t_i \cdot t_j)} \cdot t_i \right). \quad (4.9)$$

The representative map aggregates a sum of the weighted and original maps to enhance the valuable information. The proposed attention ignores the convolutional block at the initial stage to focus on computational efficiency and parameters without compromising enhancer performance. The strength of this module is to capture the most valuable elements based on the local area to improve the model performance. Each facial emotion has a unique characteristic. This property is an important element for recognizing expressions. A dimension representation block focuses on the feature characteristics and expresses a relation between elements through positional information.

Connection Module. The proposed facial expression classification needs this module to extract represented features and bridges between the attentive modules. It also employs a residual technique to retain previous features and use them as fusion information. The aggregation is utilized to combine two extracted information using element-wise addition. Additionally, It applies a bottleneck module by decreasing the channel size at the first operation, which aims to save a number of parameters, as shown in Fig. 4.1. To optimize the training process, it uses Leaky ReLU, dropout, and batch normalization techniques. It also applies a feature merging at the end of the representation module, combining it with the feature map from past attention. The fusion map keeps the substance of the previous enhancement module by comprehensively learning. It also minimizes the loss of information that is caused by the redundant operation and vanishing gradient.

4.2 Classifier Module

The classifier module plays a vital role in predicting the emotion class labels by generating prediction probabilities. The highest score is taken as a prediction result

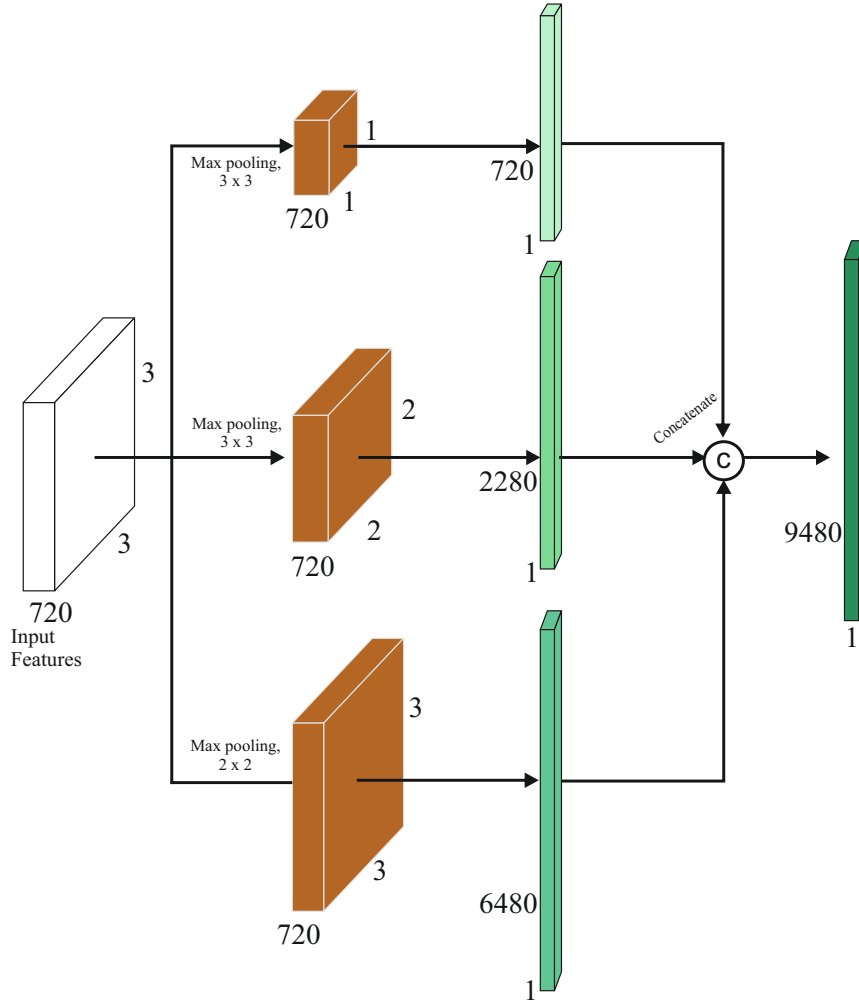


FIGURE 4.6: The modified Spatial Pyramid Pooling (SPP).

based on the score of each expression. The proposed module uses the Softmax classifier to obtain a multinomial distribution of seven classes. It normalizes the logit score of the class to be probability values, where the total is one. In order to generate vector score, it utilizes Spatial Pyramid Pooling (SPP) by modifying the pooling combination as shown in Fig. 4.6. Three adaptive pooling windows by an adjusted stride ($Pool_1$, $Pool_2$, and $Pool_3$) are employed to minimize feature loss. It produces a pyramid map representing multiple receptive fields. The SPP-modified can be written as:

$$SPP = [Fl(Pool_1(x_i)), Fl(Pool_2(x_i)), Fl(Pool_3(x_i))], \quad (4.10)$$

where Flatten $Fl(\cdot)$ is applied to convert tensor maps into a 1-dimensional array. The SPP deforms the extracted specific features and aggregates them to avert the early cropping of essential elements. Hence, a concatenate module combines all information from all three branches to enrich interest information and transfers it to

the classifier stage.

4.3 Implementation Setup and Dataset Configuration

This facial expression network was simulated in the Keras framework that uses the optimal parameter and configuration. The training stage was conducted using A GTX 1080Ti to accelerate the process. The speed test in the inference phase uses a main CPU with an Intel Core i5-6600 processor, @3.30GHz, RAM of 8GB without a graphic accelerator. In order to assess the error of prediction, Categorical Cross-Entropy loss is employed using the epsilon parameter of 10^{-7} . The proposed module utilizes Adam (Adaptive moment estimation) as an optimizer in the training phase.

The network was trained at a 10^{-4} learning rate in the starting stage. It will be updated when the performance does not improve in 20 epochs, multiplying by 0.75. The variation in the number of images from the used datasets applies to different epochs and batch sizes for each benchmark. We apply 500 epochs for the FER-2013 dataset and 200 epochs for CK+, JAFFE, and KDEF datasets. The CK+ and KDEF apply mini-batch sizes of 16, while JAFFE and FER-2013 use 8 and 4, respectively.

The proposed model was trained and evaluated on FER-2013, CK+, JAFFE, and KDEF. FER-2013 consists of 28,709 instances in the training set, 3,589 in the validation set (PublicTest), and 3,589 in the testing set (PrivateTest) based on a Facial Expression Recognition Challenge in the Kaggle competition. Additionally, we split 75% for the training and 25% for the testing set on CK+, JAFFE, and KDEF datasets in the standard category. On the other hand, it also applies 10-fold cross-validation to compare with a regular proportion. We apply initial processing in the CK+, JAFFE, and KDEF datasets using face detection to obtain the facial patch. It can help the model eliminate the background noise to increase the performance.

Furthermore, cropped image from CK+ and JAFFE is resized into 64×64 , while the KDEF demands 48×48 pixels. In order to enrich the instances and avoid overfitting in the training phase, it applies the augmentation technique on CK+, JAFFE, and KDEF datasets. In contrast, the FER-2013 dataset did not apply this method because it has many images. Brightness, contrast, and horizontal flipping are utilized for CK+ and JAFFE. Additionally, This method is applied that is more varied

on the KDEF dataset, such as illuminance, color distortion, rotation, and flip transformation. This dataset is knowledge for the real-case application provided with multi-pose instances.

4.4 Experimental Results

The proposed architecture is analyzed in several ablative experiments to assess each module’s strength. This section also explains network performance evaluation and compares it to the other methods on the facial expression datasets.

TABLE 4.1: Comparison of efficient backbone with other networks.

Network	Num. of parameter	Accuracy (%)	Computational Complexity (GFLOPS)
VGG13	9,419,207	72.47	1.02
VGG16	14,734,023	72.23	1.41
ResNet-18	11,192,647	62.89	0.06
ResNet-34	21,311,943	62.94	0.12
Proposed	5,506,231	72.47	0.80

4.4.1 Backbone Analysis

A CNN-based architecture tends to have problems when it requires implementation in real-time applications. It is because the model generates huge parameters and high computational requirements. Hence, a proposed backbone is introduced to rapidly extract essential features with partial transfer at particular blocks. As shown in Table 4.1, the proposed technique generated the least number of learnable weights. The accuracy is evaluated on the FER-2013 benchmark. Even though it has the same performance as the VGG-13, it generates a smaller 1.7 times parameter. On the other hand, ResNet-18 and ResNet-34 reach low performance and produce many parameters.

The analysis of the partial transfer illustrates that this module is effectively applied to multiple blocks, as shown in Table 4.2. This module is sequentially inserted from the back of this network without changing the structure of the convolutional layers. Although it obtains high accuracy when not installed partial transfer, it generates 6,274,087 parameters. In addition, a backbone with two partial transfer modules obtains the lowest parameters, but it achieves lower accuracy than using three

modules. Finally, we select three partial transfers installed on the proposed backbone because it obtains optimal results.

TABLE 4.2: Ablative study of transferred partial module.

Num. of transfer layers	Num. of parameter	Accuracy (%)
0	6,274,087	72.50
1	5,584,007	72.47
2	5,492,295	72.33
3	5,506,231	72.47
4	5,546,579	71.75

TABLE 4.3: Model analysis of each proposed module.

Exp	EPT module	SPP module	Global attention	Channel attention	Dimension attention	Acc (%)	Parameters (M)	GFLOPS	Speed (FPS)
1	✓					72.47	5.51	0.797	80.37
2	✓	✓				72.97	5.57	0.797	79.43
3	✓	✓	✓			73.50	6.18	0.807	74.45
4	✓	✓	✓	✓		73.89	6.38	0.832	70.24
5	✓	✓	✓	✓	✓	74.17	6.58	0.835	69.18

4.4.2 Proposed Model Analysis

This ablation study is conducted in the same training setting, besides particular changes to each module. Table 4.3 illustrates that the proposed module has a positive impact by increasing the accuracy. It uses the FER-2013 dataset to examine the accuracy. Firstly, the baseline structure only applies an efficient backbone with the softmax classifier that achieves the performance of 72.47%. In addition, it generates 5.51M parameters. Secondly, the flattened layer in the extractor module is replaced with the SPP-modified module that improves the accuracy by 0.5% and adds 60K parameters. Thirdly, the global attention module is inserted after the backbone module achieves 73.5% accuracy and produces 6.18M parameters.

Fourthly, it inserts a channel attention network before the SPP-modified module. It shows improving the accuracy, parameters, and computations. Finally, the dimension attention module improves the network performance by 0.28% by adding 200,000 parameters. In addition, it evaluates each module’s speed by integrating them with an LWFCPU face detector (Putro, Nguyen, and Jo, 2020). A global attention module significantly reduces the speed by 4.98 FPS. In contrast, the dimension attention module only decreased by 1.06 FPS. Based on those results, each attention

module can enhance the backbone performance and maintain its efficiency to avoid a significant reduction.

TABLE 4.4: Formation analysis of proposed attention modules.

Module	Experiment							
	S-1	S-2	S-3	S-4	S-5	S-6	P	P*
Global att.	1	1	2	3	2	3	-	-
Channel att.	2	3	1	1	3	2	-	-
Dimension att.	3	2	3	2	1	1	-	-
Acc (%)	74.17	74.11	73.59	73.61	74.11	73.81	73.84	73.67

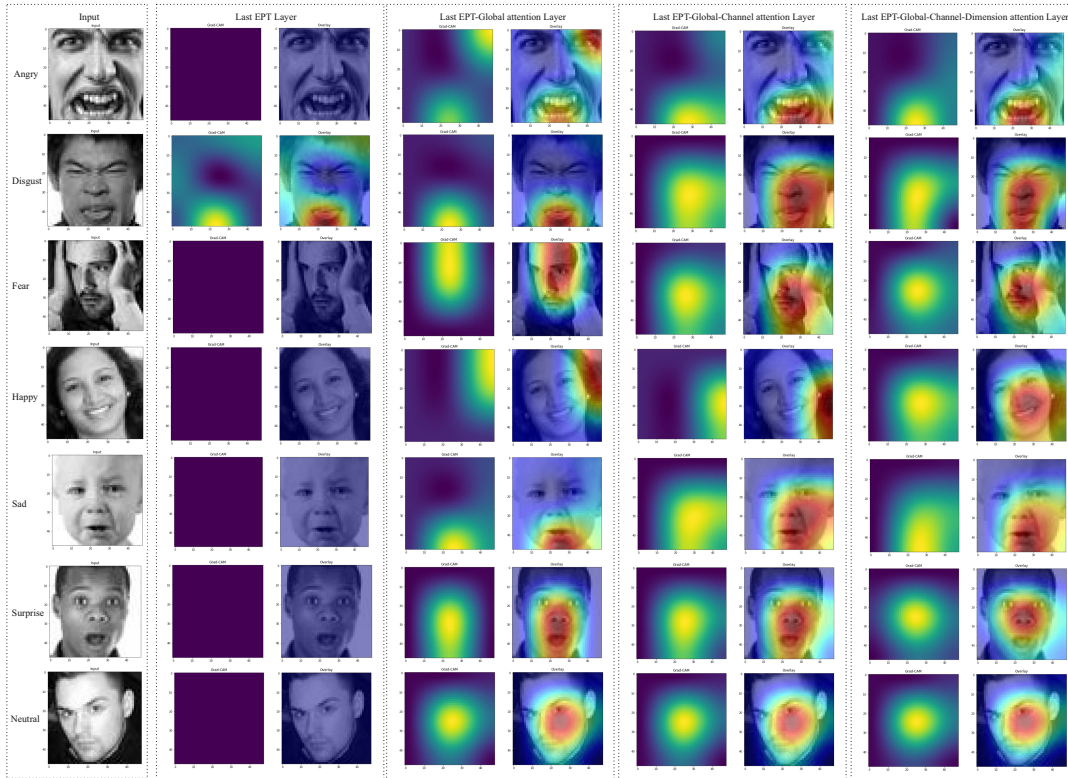


FIGURE 4.7: The heatmap visualizes the feature attention at each representation module.

Furthermore, the formation of the attention model is investigated with different structures and order, as illustrated in Table 4.4. S-(·) is sequential formations in several experiments. P is a parallel arrangement by summing all elements of both maps on the same coordinate and location. P* applies to concatenate technique. Low accuracy is achieved when channel attention is installed on the top formation. Because this block plays a role in capturing useful contextual features, it can effectively insert after the global attention module. Besides, the parallel configurations also does not acquire the best accuracy. In conclusion, the first series formation (global-channel-dimension) obtained excellent performance compared to all the configurations.

TABLE 4.5: Comparison of the proposed model with to other methods on different datasets.

Datasets	Model	Accuracy (%)
FER-2013	Multi-scale CNN	72.82
	SNNs (Hayale, Negi, and Mahoor, 2021)	73.00
	Single MLCNN (Nguyen et al., 2019)	73.03
	Hybrid CNN-SIFT aggregator	73.40
	Ensemble MLCNNs (Nguyen et al., 2019)	74.09
	AM-Net (Gan et al., 2020)	75.82
	Proposed model	74.17
CK+	MA-Net (Gan et al., 2020)	96.28
	CCRNet (Xi et al., 2021)	98.14
	ExpNet+Fusion (Otherdout et al., 2020)	98.40
	AM-Net (Li et al., 2020)	98.68
	MGLN-GRU	99.08
	Baseline+STCAM (Chen et al., 2020)	99.08
	Proposed model	99.18
	Proposed model (K-Fold)	99.17
JAFPE	Li et al.	91.80
	DCMA-CNN	94.75
	CO-CLS FER	95.31
	Wang et al.	95.70
	Hamester et al.	95.80
	AM-Net (Li et al., 2020)	98.52
	Proposed model	98.75
	Proposed model (K-Fold)	98.82
KDEF	CRC	90.24
	PCRC	90.71
	O-FER	91.42
	CCFN	91.60
	Multi-Model fusion	93.42
	DFSD-LDA (Palaniswamy and Suchitra, 2019)	95.06
	Akhand et al., 2021	96.51
	Proposed model	97.12
	Proposed model (K-Fold)	97.10

Additionally, it observes the class activation map to display the attention area affected by each representation module. This method utilizes GRAD-CAM (Selvaraju et al., 2017) to visualize the crucial area containing the valuable components that are used to predict the emotion. Fig. 4.7 shows that the dimension attentive module increases the intensity of attention on local features (mouth, cheeks, eyes, and nose).

4.4.3 Evaluation on Dataset

FER-2013 dataset. This benchmark contains 35,887 gray images with 48×48 pixels resolutions. It covers the basics of seven emotions: neutral, anger, sad, fear, surprise, happy, and disgust. Moreover, the FER-2013 dataset is a challenging dataset that provides different views, characters, and ages. It contains several invalid labels that make it difficult for models to acquire high accuracy.

The proposed model reached a 74.17% accuracy that is lower than AM-NET (Li et al., 2020) as a leading competitor. However, it is superior than the Ensemble ML-CNNs (Nguyen et al., 2019) model, as illustrated in Table 4.5. To analyze in detail the performance, it shows a confusion matrix in Fig. 4.8 (a). A "Happy" expression achieves the highest accuracy. The model most mispredicted the "Sad" category as the "Neutral" emotion.

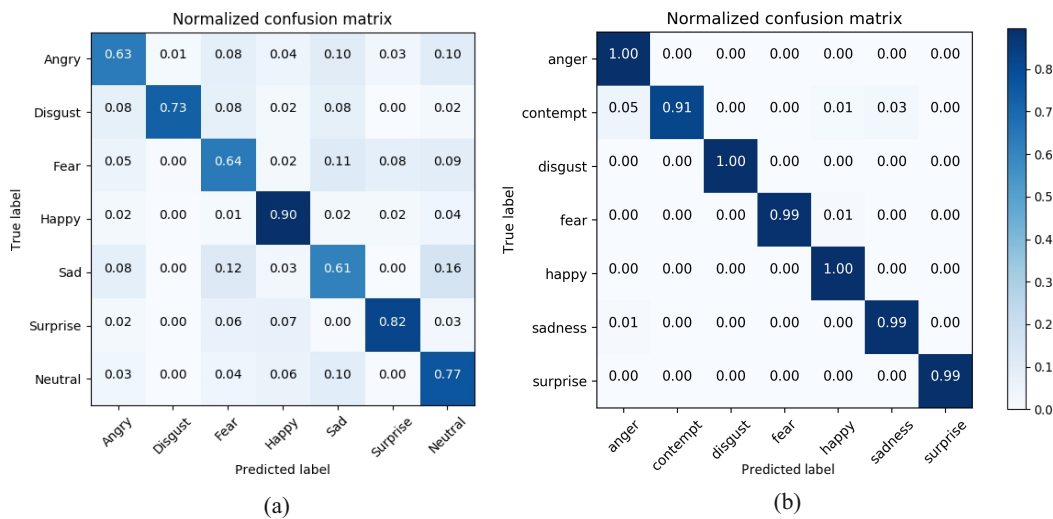


FIGURE 4.8: A comprehensive evaluation of the proposed model in confusion matrix on (a) FER-2013 and (b) CK+ datasets.

CK+ dataset. The public dataset contains 123 subjects with 593 sequential images, capturing the last three frames to produce 981 pictures. The proposed network explored this dataset containing seven expressions: sadness, anger, disgust, surprise, happy, contempt, and fear. The proposed network is examined on the standard and 10-fold evaluations to obtain a fair performance comparison. The common evaluation set is 25% of all total images.

The proposed model achieves 99.18% and 99.17% accuracy on standard and 10-fold evaluation, respectively. Meanwhile, it outperforms STCAM (Chen et al., 2020) as a leading method. The AM-NET (Li et al., 2020) also obtains lower accuracy than

our model. Fig. 4.8 (b) shows that "anger," "disgust," and "happy." obtained perfect accuracy in a comprehensive evaluation. On the other hand, the proposed model is weak in predicting "contempt" because a few instances are predicted as "anger," "happy," and "sadness."

JAFFE dataset. The dataset contains 213 instances, and each image has a resolution of 256×256 pixels. It also covers the basics of seven expressions: sad, angry, surprise, disgust, happy, neutral, and fear. The database was collected from the Japanese woman's face captured in a laboratory-based environment.

The proposed model reaches perfect performance when identifying the "angry," "fear," "neutral," "sad," and "surprise" emotions. Fig. 4.9 (a) illustrates that "disgust" class obtains the lowest performance. However, this does not prevent our network to achieves an excellent performance of 98.75% on the standard testing set. It also reaches 98.82% accuracy on the 10-fold evaluation set. These results slightly outperformed the AM-NET model (Li et al., 2020), where this competitor reaches 98.52% accuracy.

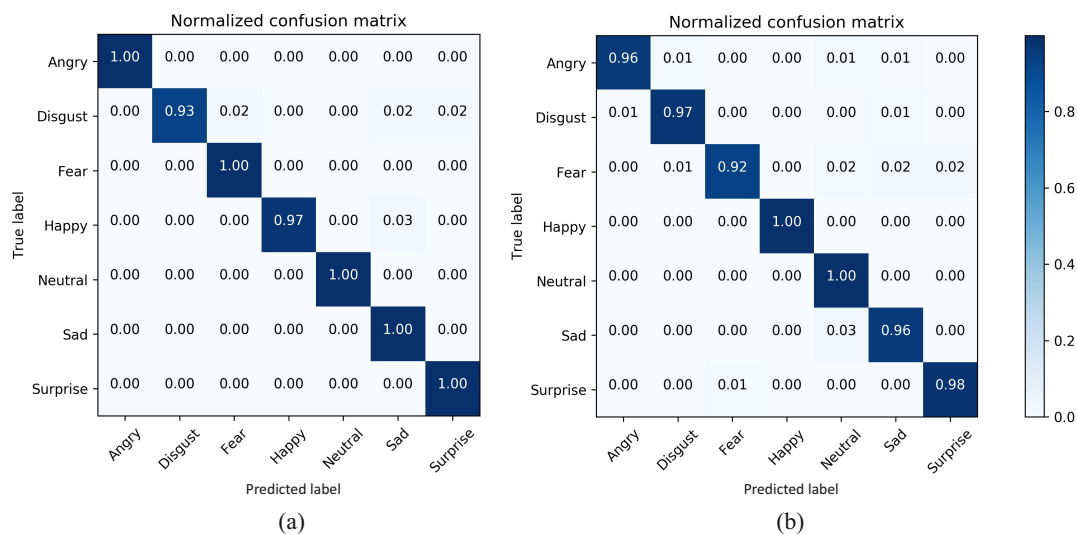


FIGURE 4.9: A comprehensive evaluation of the proposed model in confusion matrix on (a) JAFFE and (b) KDEF datasets.

KDEF dataset. This dataset provides 4,900 RGB pictures covering seven expressions: neutral, angry, happy, surprise, fear, sad, and disgust. It contains 70 subjects with five different angles of the facial pose, including straight, half right, full right, half left, and full left poses. Different facial views are the main challenge of this dataset which only presents a partial face component.

As demonstrated in Table 4.5, our model achieves 97.12% and 96.63% accuracy

TABLE 4.6: Evaluation of multi-pose facial expression recognition on KDEF dataset.

Face position	Accuracy (%)
Full left	96.03
Full right	95.48
Half left	98.33
Half right	97.89
Center	98.11

on the standard evaluation set and the 10-fold evaluation set, respectively. There is not much difference in the result when performing the two kinds of evaluation. Additionally, the model performance is superior to the DFSD-LDA (Palaniswamy and Suchitra, 2019). Fig. 4.9 (b) describes that "happy" and "neutral" emotions obtains the excellent accuracy. However, "fear" faces have false predictions of 8% against other expressions.

Furthermore, the proposed model is evaluated in realistic pose variation scenarios on the KDEF dataset. Table 4.6 illustrated that our model obtains the highest accuracy at half left position. In contrast, it obtains low performance in the full right view. Fig. 4.10 demonstrates that "fear" predicts a low true positive compared to other categories of expressions.

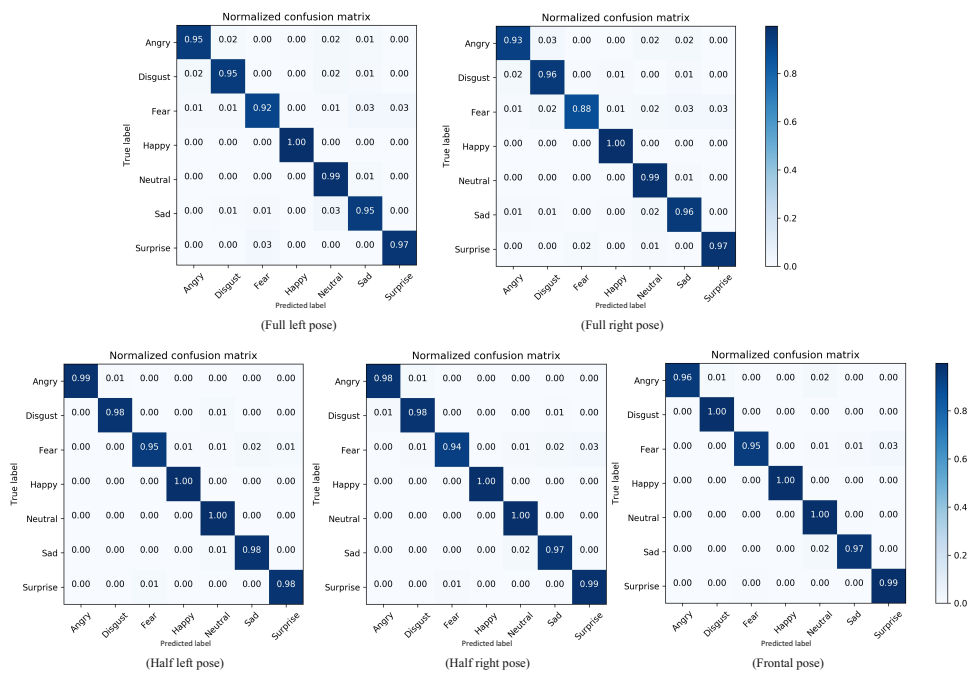


FIGURE 4.10: Confusion matrix of multi-pose evaluation on KDEF datasets.

Chapter 5

Integrated Facial Emotion Recognition

The integration module combines several cooperated systems to estimate the location of faces and identify their emotions. This work incorporates face detection and expression classification modules that can run at real-time speed on inexpensive devices, as illustrated in Fig. 5.1. This section explains the combination of the proposed modules into a working system in detail. It uses a face detector discussed in Chapter 3 and a facial expression classification described in Chapter 4.

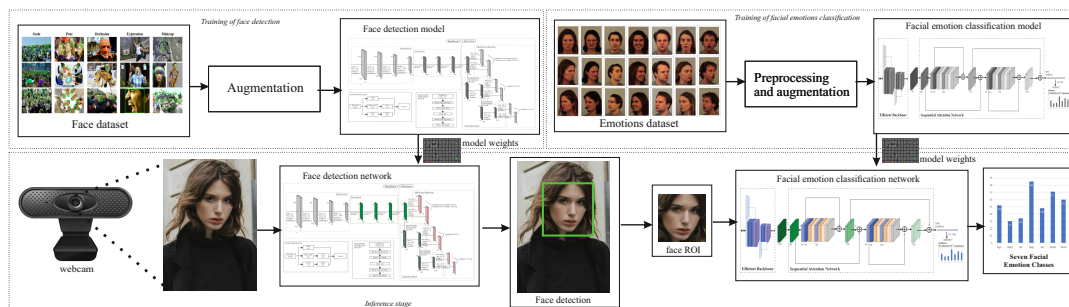


FIGURE 5.1: Integration system of facial emotion recognizer.

The proposed detector is examined in low-cost computing devices to measure the capacity of the vision method in practical applications. The primary tester uses an Intel Core I5-6600 @3.3GHz CPU with 8GB RAM. In addition, this experiment was also conducted in a Jetson Nano with a Quad-core ARM Cortex-A57 MPCore processor, NVIDIA Maxwell 128 GPU, 4GB of RAM. Both devices run Ubuntu 18.04.3 LTS operating system and simulate all proposed modules in the Pytorch framework. This experiment employs an RGB camera to obtain live video streams at 30 fps. It uses various video resolutions according to the experimental needs of each module.

The initial stage employs a face detector to localize faces from an input frame. It separates the entire background from the detected face, which assists the emotion classifier focus on the region of interest of the face. In addition, it can improve emotion detection performance which avoids background noises. Furthermore, the facial expression module classifies the facial area into seven emotion classes.

5.1 Runtime Efficiency of Face Detector

A Deep learning face detectors generally use additional high accelerators to boost the processing speed model. However, this is expensive, even though the practical application demands that it run on cheap devices. Therefore, this lightweight detector is an immediate prospect to implement on this device. The proposed face detector produces 989,832 parameters with 0.195 GFLOPS.

The detector speed was examined using CPU-based devices at video graphic array resolution (640×480 pixels). It uses a true positive rate that describes the detection precision with a maximum of 1,000 false positives on the FDDB dataset, as shown in Table 5.1. It applies a 0.05 confidence threshold and a 0.3 Non-Maximum Suppression to generate the final prediction.

TABLE 5.1: Runtime efficiency compared to different face detectors on CPU.

Detector	CPU Device	TPR(%)	FPS on VGA
ACF	Intel I7-3770 @3.40GHz	85.20	20
CasCNN	Intel E5-2620 @2.00GHz	85.70	14
FaceCraft	Intel 4770K @3.50GHz	90.80	10
STN	Intel I7-4770K @3.50GHz	91.50	10
MTCNN (Zhang et al., 2016)	N/A	94.40	16
DCFPN (Zhang et al., 2018)	Intel E5-2660v3 @2.60GHz	95.40	30
FaceBoxes (Zhang et al., 2019)	Intel E5-2660v3 @2.60GHz	96.50	28
FaceBoxes (Zhang et al., 2019)	Intel I5-6600 @3.30GHz	96.50	39
INCEPTION V4 (Szegedy et al., 2017)	Intel I5-6600 @3.30GHz	95.16	2
RetinaFace-Mobile (Deng et al., 2019)	Intel I5-6600 @3.30GHz	97.26	19
Proposed	Intel I5-6600 @3.30GHz	97.00	53
Proposed with six mini-inception	Intel I5-6600 @3.30GHz	97.36	44

This result shows that the our detector operates at a processing speed of 53 FPS. It is faster than FaceBoxes detector, as the latest competitor. It also outperforms the INCEPTION V4 on an Intel I5-6600 CPU. Although the RetinaFace-mobile version achieves high accuracy, it is 34 FPS slower than the proposed face detector. On the

other hand, a six mini-inception version achieves superior accuracy and runs faster than the competitor on the same device.

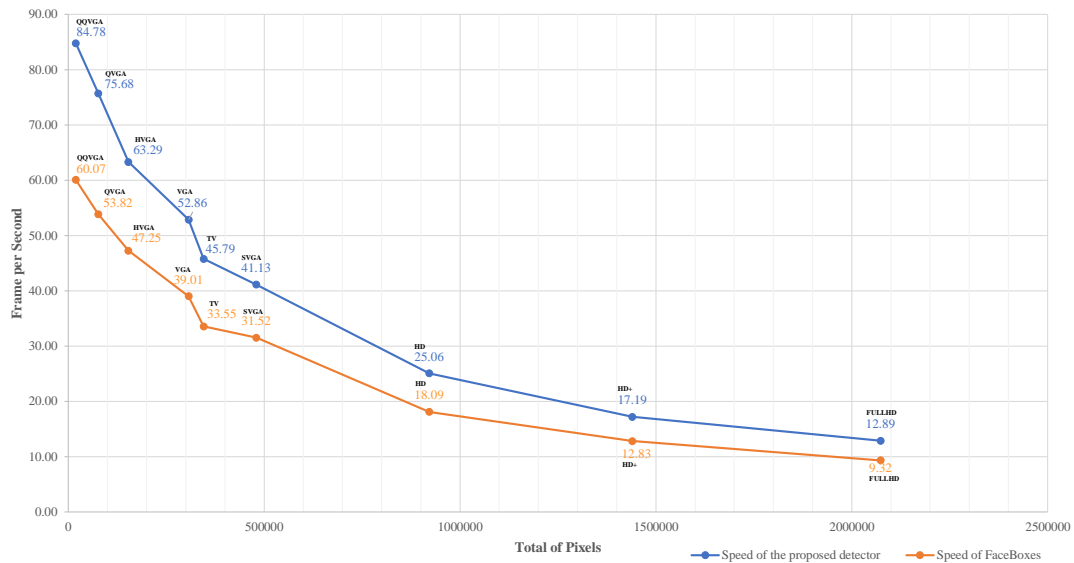


FIGURE 5.2: The proposed model speed is compared with FaceBoxes at different input resolutions.

Figure 5.2 compares the speed of the proposed detector with FaceBoxes in different input video sizes. It shows that competitors run slower on all resolution sizes. The proposed detector achieves 25 FPS at HD resolution, which indicates the detector can operate in real-time at large input sizes.

TABLE 5.2: Comparison of model speed in real-time application on the low-cost and an edge device

Devices	RAM (GB)	Proposed Detector	RetinaFace mobile0.25	FaceBoxes
PC desktop Intel Core I5-6600	8	52.86	18.67	39.00
Notebook AMD A6-1450 224	4	10.10	2.39	8.44
Lattepanda Intel Cherry Trail	4	8.76	1.51	8.36
Raspberry pi 3B Broadcom BCM2837	2	2.15	0.19	1.37
Jetson Nano NVIDIA Maxwell 128 GPU	4	34.97	11.69	30.66

5.2 Real-time Application of Face Detector on Low-cost Devices

The real-world scenario challenges a computer vision method to implement in practical applications. The accuracy and processing time are crucial trade-offs for real-time applications. In addition, practical applications demand a method to run on low-cost devices instead of applying it on expensive devices.

The proposed detector is examined the processing speed on several low-cost and edge devices, as presented in Table 5.2. The detectors are tested from a webcam in 1,000 frames. The experiment demonstrates that each detector is operated on several low-cost devices. The retail price of each hardware is not higher than 300 USD. The proposed detector can work faster than FaceBoxes and RetinaFace-mobile on all devices with VGA resolution. Moreover, competitors are slow to operate on Notebook, Lattepanda, and Raspberry-pi. These results can be concluded that our detector has superior speed when implemented on low-cost devices.

Different processor size affects the speed of a CNN-based model to be able to run fast on a device. A slow processing rate is performed on a raspberry-pi. It is caused that a CPU in this device has a low execution rate, which discourages the computational module from operating fast. When tested on low-cost devices, our detector needs less operation time and can operate at a reasonable rate. This architecture generates low computation, small memory usage, and lightweight parameters by employing fewer filters required at each convolutional block.

The visualization result presents that the proposed detector can accurately localize faces of various sizes. As shown in Figure 5.3 (c), the small face does not weaken the detector to find its location. The partial occlusion face in Figure 5.3 (b) can be accurately detected by the proposed detector. In addition, the detector can predict the location of faces in extreme position challenges, as presented in Figure 5.3 (a).



FIGURE 5.3: Visualization of the face detection results on live streams video at VGA (a), HD (b), and Full HD (c) resolution.

TABLE 5.3: Comparison of the integrated model speed on Intel Core i5 CPU.

Evaluation	Model			
	Single MLCNN	Ens. MLCNN	AM-Net	Proposed
Accuracy on FER-2013 (%)	73.03	74.09	75.82	74.17
Parameters (M)	20.79	92.82	24.90	6.58
GFLOPS	1.53	4.64	2.98	0.84
Model speed (FPS)	65.25	30.49	41.25	90.03
Integrated speed (FPS)	37.96	22.81	28.35	45.24

5.3 Runtime Efficiency of Facial Emotion Recognizer

A CNN-based model can smoothly work in real-time using a graphical accelerator, but this device is expensive, whereas practical applications require it to be implemented on low-cost devices. The facial expression model is designed to run fast on a cheap device. It generates 6,578,243 learnable parameters with 0.835 Giga Floating Points Operations (GFLOPs). To implement it in real-time applications, the proposed facial expression model described in Chapter 4 was integrated with the face detection discussed in Chapter 3. The model integration is examined its efficiency on live stream video using a webcam.

Face detection works at the first stage to determine the facial location. It then crops the area to obtain the region of interest. According to the input dimension of the facial expression model, a cropped image is rescaled to 48×48 . It then enters the resized patch into the facial expression model. The efficiency experiments were conducted on several low-cost devices, including the PC, a Laptop, a Notebook, an embedded device, and an edge device. Table 5.3 shows that the proposed facial expression network has a speed of 90.03 FPS on a CPU with a Core i5 processor. This result shows that it is the fastest model. Although the model performance differs by 1.65% from AM-Net, the computation complexity and parameters are more efficient.

Based on these results, it has two significant benefits of the proposed network compared to another model. Firstly, this model uses an EPT backbone to help the integrated model rapidly operate on low-cost computing devices, supporting low computational cost. In comparison, another competitor commonly utilized deep CNN architecture to achieve excellent performance.

MLCNN employed the ensemble structure to fuse texture information with complex features in the last stage. On the other hand, an AM-NET model applied a twin

feature extractor and used a dense connections block that produced more than 20M parameters. They use abundant filter channels, which generate many numbers of operations. Therefore, both models depend on an additional accelerator to execute parallel computing and boost speed in the inference phase.

The proposed facial expression module applies the partial transfer approach to compress the redundant operation in the convolution process. This method reduces the channel size of the filter at the initial layer, which can suppress the usage of many parameters.

Secondly, this work promotes a sequential attention module that uses a cascade structure to effectively increase the backbone performance. It comprehensively highlights the specific facial elements that improve the feature map quality. Global attention is applied to upgrade the global contextual information and relationship between each feature. Then, the channel and dimension attention blocks sequentially enhance the potency of the essential elements based on channel and positional mapping information.

This cascade attention network gradually improves EPT output to achieve competitive accuracy with the MLCNN and AM-NET. The proposed modules do not significantly decrease integrated model speed in a real-time application. Thus, this architecture focus on network efficiency, which does not compromise its performance. The facial expression network allocates less memory and uses low computation power, which is able to operate at a reasonable rate on low-cost devices.

5.4 Real-time Application of Face Emotion Recognizer on Low-cost Devices

Nowadays, a robot has been widely employed in public places to serve humans. It requires an emotion recognizer system to identify the person's facial expressions. Each emotion helps interaction activities by representing the response from the user. Thus, the facial emotion method is really needed by an interactive robot to sense a person's expressions, and then it will be an input signal to respond to the user's feelings correctly. In addition, the capability of the human-robot interaction can be increased by implementing the emotion recognizer in real-time processing speed.



FIGURE 5.4: Visualization of detection results in the integrated system on a live streams video.

The proposed emotion recognizer can accurately localize the face area and classify the seven basic facial expressions. The learning process builds characteristics

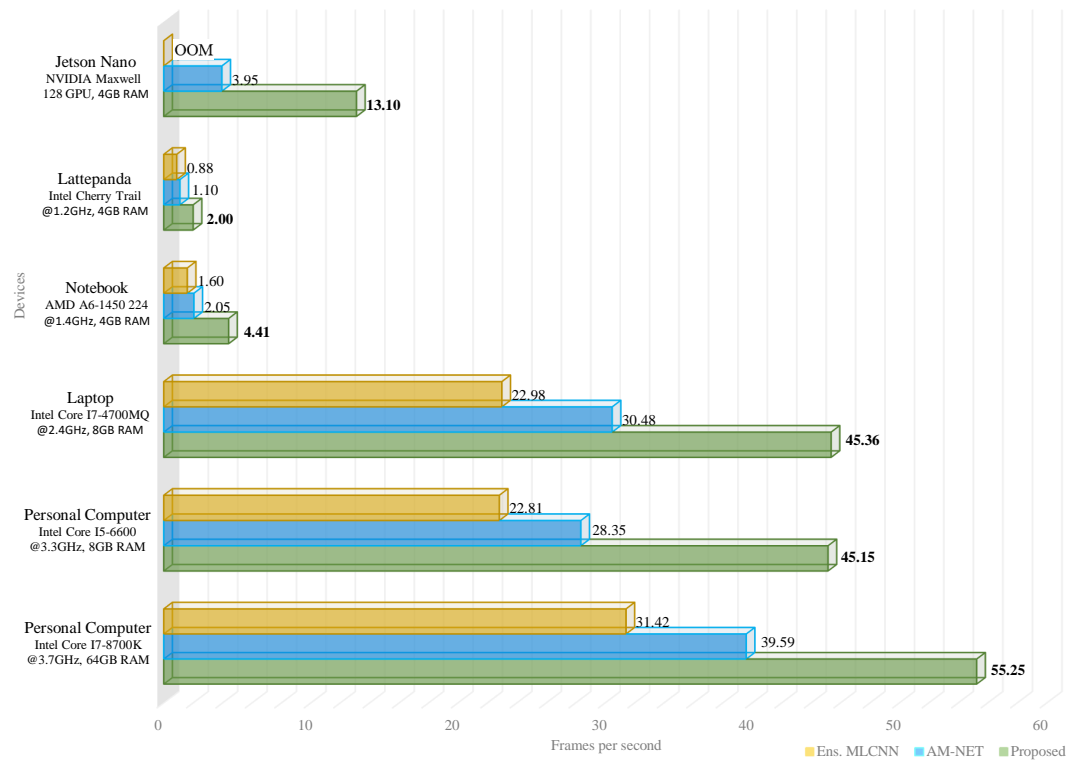


FIGURE 5.5: Comparison of integrated model speed implemented on low-cost devices.

of the training data to discriminate the interest element. The proposed model takes facial information from KDEP as real-application detector knowledge. This dataset covers multi-views that are more representative in real-world scenarios. Fig. 5.4 illustrates that the emotion recognizer accurately identifies a human facial expression. This result presents that this system can effectively estimate a facial emotion on a human face. It can recognize each facial expression in multiple persons. The visualization proves that the proposed network is acceptable as a real-time model for supporting the human-robot interaction.

The robotics needs a vision model to execute fast and run at real-time speed. It suppresses the delay time of all operations to boost the processing rate to work without stuttering. Additionally, a robot usually uses low-cost devices to acquire information from sensors and synchronize it with actuators. Accordingly, the application of a method in an embedded device is needed to increase the capability of the vision-based system.

The model efficiency is tested on other low-cost devices and compared with other competitors. Fig 5.5 presents that the integrated model reaches speeds of 2, 4.41, 13.10, 45.15, 45.36, and 55.25 FPS on Lattepannda, Notebook, Jetson Nano, Core i5

PC, Core i7 Laptop, and Core i7 PC, respectively. These devices have been widely used in robotics, where each device is directly connected to a webcam to implement the facial emotion recognizer by live streaming video.

The speed results demonstrate that our emotion recognizer is faster than other methods. This experiment also integrates the competitor's model with the proposed face detector. Even though the integration of the proposed model operates slowly on a Lattepanda, this device has a low clock rate. However, the proposed emotion recognizer is more acceptable than other methods to implement in this device for identifying a person's facial expressions.

Another experiment was conducted on Jetson Nano as an embedded system with a small GPU. This result showed that the speed of the ensemble model occurs out of memory (OOM). A mini accelerator could not accommodate the competitor's excessive memory and computational usage. Meanwhile, AM-NET only obtains 3.95 FPS, restraining real-time operating on this edge device.

This observation presents that the competitors need much more processing time to be implemented an embedded device for classifying the expressions. Eventually, the proposed models establish a cognitive detector to identify a human facial expression used for emotional intelligence. The proposed architectures effectively suppress computational cost and memory usage, which affects increasing model speed. Moreover, the proposed facial emotion recognizer can run faster than other competitors, especially execution on low-cost devices.

Chapter 6

Conclusion

6.1 Conclusions

This work focus on an integrated facial emotion recognition system that incorporates the efficient architecture of face detector and facial expression networks. The proposed system can robustly recognize facial emotion, even with a complex background. This integrated model can smoothly operate in real-time processing data speed on inexpensive hardware. An entire architecture is discussed alongside the comprehensive explanation correlated to each proposed module.

This thesis discusses an efficient CNN-based architecture on a multi-detection face detector model in the first section. The backbone robustly discriminates the distinctive features and rapidly predicts the multiple faces scales. According to the experiments, this strategy improves the average precision score on all benchmarks containing various challenges.

The high-performance and efficient face detector is designed using a light convolutional neural network that predicts the location of faces on several scales. A shrink block rapidly extracts features and reduces the dimension of the feature map. A stem module employs the mini-inception module that enriches features by raising the receptive field area. Therefore, a combination of shrink and stem modules can deliver excellent detection performances because these modules can accurately distinguish between facial and background features.

The pyramidal feature hierarchy is employed to avoid extra computation and assigns the low, medium, and high layers to predict face location using multiple anchors. This approach can increase the detector's efficiency by reducing additional

operation and memory usage compared with FPN architecture. The module is divided into three levels for the detection layer by estimating the face location into three scales according to the prediction map dimension.

This proposed detector applies an anchor technique to initialize the predicted bounding boxes. It uses a square ratio to occupy different prediction layers. In addition, the anchor scale variation helps the detector to reduce the gap in variation face sizes. Our face detection applies the anchor densification technique to improve precision in the small face category.

The balancing loss and training strategy enhance prediction precision in the training stage. These methods have no impact on reducing the speed detector in the inference phase. The proposed detector applies multi-boxes loss that integrates regression and classification loss by assigning it to each anchor. Augmentation techniques and gradual learning rates help the model comprehensively learn variations in facial features to generate robust models.

According to the experiments, each proposed module improves detector performance by avoiding significant parameters and computational overhead. Light backbone with shrink and stem module maintains detector efficiency without weakening detection performance. Multi-level detection increases the 5.9% true positive rate without significantly reducing processing time. The proposed face detector achieves 97% average precision that outperforms other low-cost face detectors. Additionally, the network efficiency outperforms other detectors by achieving 53 FPS on a Core i5 CPU and 35 FPS on a Jetson Nano.

For the second section, this thesis focus on classifying seven basic emotion classes. An efficient deep learning model is designed using the EPT module to rapidly discriminate between interest and trivial facial components. It splits the input features map into two, extracts one of its parts, and combines it to enrich the information. This module extracts a partially input map with less computational complexity than the common block. It also keeps the performance by fusion in different frequency features.

This facial expression model also employs a cascade attention network to capture essential features and selectively boost the intensity. The series module consists of three attentive modules to highlight global and local feature representation. The

global attention module can enhance the global context features, while channel and dimension attention modules are concentrated on the local attention features.

Furthermore, a classifier module is assigned to estimate the predicted probability and applies SPP-modified to generate the raw vector. The SPP employs three adaptive pooling windows to prevent the loss of essential information. This module generates pyramid feature maps with multi-dimensions in different receptive fields. Additionally, a construction module helps to extract represented features and bridges all attention networks. It applies a transfer connection to retain previous information by fusing features at different levels.

According to the experiments, the EPT model retains around 72% performance on the FER-2013 dataset and outperforms other baseline networks. A combination module achieves competitive accuracy of 74.17% with 6.58M parameters and 0.835 GFLOPS. The module reaches the processing speed of 90.03 FPS on a Core i5 CPU. This result indicates that the proposed face expression model is more efficient and faster than other competitors.

An integrated model combines face detection and facial expression classification module to detect human facial emotion. It implements face detection in the beginning process to generate regions of interest. Then, the facial expression module is applied to the RoI to predict the seven basic emotions. In order to measure the efficiency system, a two-stage network is tested by live streams video on low-cost computing devices.

According to the experiments, the integration of the proposed module is faster than other integrated modules. It achieves 2, 4.41, 13.10, 45.15, 45.36, and 55.25 FPS on Lattenda, Notebook, Jetson Nano, Core I5 PC, Core I7 Laptop, and Core I7 PC, respectively. The proposed system provides a reliable model that could work in real-time and is suitable for real-world applications.

6.2 Future Works

In this thesis, the proposed system is focused on designing the architecture to improve efficiency without compromising its performance. On the other hand, the loss

function can increase the effectiveness of the training without reducing the inference speed. Therefore, the future work of this study is to explore error evaluation to enhance prediction accuracy.

Another important aspect is a feature extractor that can be upgraded with a transformer approach. This model has shown excellent performance in obtaining distinctive features based on self-attention. This method can be applied with a more efficient design and increase predictive precision, encouraging a model to be implemented on mobile devices.

Furthermore, the application of real-world scenarios is the actual challenge of a deep learning method. This emotion recognizer will be implemented on an assistive robot to examine the reliability by quantifying customer satisfaction using their expressions.

Appendix A

Publications

A.1 Journal

1. Muhamad Dwisnanto Putro, and Kang-Hyun Jo, High Performance and Efficient Real-time Face Detector on CPU Based on Convolutional Neural Network, *IEEE Transactions on Industrial Informatics*, 2021.
2. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo, An Efficient Face Detector on a CPU Using Dual-Camera Sensors for Intelligent Surveillance Systems, *IEEE Sensors*, 2022.
3. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo, A Fast CPU Real-time Facial Expression Detector using Sequential Attention Network for Human-robot Interaction, *IEEE Transactions on Industrial Informatics*, 2022.
4. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Facemask Wearing Alert System Based on Simple Architecture with Low-Computing Devices, *IEEE Access*, 2022.
5. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Driver Behaviors Recognizer Based on Light-weight Architecture and Attention Mechanism, *IEEE Access*, 2022. (under review process).
6. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Lightweight Eye Status Detection Architecture for Drowsiness Warning with Low-computing Devices, *IEEE Transactions on Industrial Informatics*, 2022. (under review process)

A.2 Conference

1. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, Adri Priadana, and Kang-Hyun Jo, A Faster Real-time Face Detector Support Smart Digital Advertising on Low-cost Computing Device, AIM 2022, Royton Sapporo, Sapporo, Japan, July 11, 2022.
2. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo, A CPU-based Pedestrian Detector using Deep Learning for Intelligent Surveillance Systems, ICIT 2022, Shanghai, China, August 22, 2022.
3. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, Adri Priadana, and Kang-Hyun Jo, Fast Person Detector with Efficient Multi-level Contextual Block for Supporting Assistive Robot, ICPS 2022, University of Warwick in Coventry, United Kingdom, May 24, 2022.
4. Adri Priadana, Muhamad Dwisnanto Putro, Kang-Hyun Jo, An Efficient Face Gender Detector on a CPU with Multi-Perspective Convolution, ASCC 2022, Jeju, Korea, May 4, 2022.
5. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, Xuan-Thuy Vo, and Kang-Hyun Jo, Convolutional Neural Network Design for Eye Detection under Low-illumination, IW-FCV 2022, Hiroshima, Japan, Feb 21, 2022.
6. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo, A Fast Real-time Facial Expression Classifier Deep Learning-based for Human-robot Interaction, ICCAS2021, Jeju, Korea, Oct 12, 2021.
7. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Light-weight Convolutional Neural Network for Distracted Driver Classification, IECON 2021, Toronto, Canada, Oct 13, 2021.
8. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Distracted Driver Recognizer with Simple and Efficient Convolutional Neural Network for Real-time System, ICCAS2021, Jeju, Korea, Oct 12, 2021.
9. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo, Efficient Face Detector Using Spatial Attention Module in Real-Time Application on an Edge Device, ICIC2021, Shenzhen, Guangdong, China, Aug 12, 2021.

10. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, Xuan-Thuy Vo, and Kang-Hyun Jo, Triple Detector based on Feature Pyramid Network for License Plate Detection and Recognition System in Unusual Conditions, ISIE 2021, Miyako Messe, Kyoto, Japan, Jun 20, 2021.
11. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Eye State Recognizer Using Light-weight Architecture for Drowsiness Warning, ACIIDS 2021, Phuket, Thailand, Apr 7, 2021.
12. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo, Real-time Multi-view Face Masks Detector on Edge Device For Supporting Service Robots in the COVID-19 Pandemic, ACIIDS 2021, Phuket, Thailand, Apr 7, 2021.
13. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo, SGC-Net: Spatial-Global Context Attention Network for Real-time Facial Expression Recognition, IWIS 2020, Ulsan, Korea, Dec 13, 2020.
14. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Proposed Light-weight Convolutional Neural Networks for Real-time Hand Gesture Detector, IWIS 2020, Ulsan, Korea, Dec 13, 2020.
15. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Human Eye Detector with Light-weight and Efficient Convolutional Neural Network, ICCCI 2020, Da Nang, Viet Nam, Nov 30, 2020.
16. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Eyes Status Detector Based on Light-weight Convolutional Neural Networks supporting for Drowsiness Detection System, IECON 2020, Marina Bay Sands Expo and Convention Centre, Singapore, Oct 18, 2020.
17. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo, A Dual Attention Module for Real-time Facial Expression Recognition, IECON 2020, Marina Bay Sands Expo and Convention Centre, Singapore, Oct 18, 2020.
18. Duy-Linh Nguyen, Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Hand Detector based on Efficient and Lightweight Convolutional Neural Network, IC-CAS 2020, Busan, Korea, Oct 13, 2020.

19. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo, Fast Eye Detector Using CPU Based Lightweight Convolutional Neural Network, IC-CAS 2020, Busan, Korea, Oct 13, 2020.
20. Muhamad Dwisnanto Putro, and Kang-Hyun Jo, Fast Face-CPU: A Real-time Fast Face Detector on CPU Using Deep Learning, ISIE 2020, Delft, Netherland, Jun 17, 2020.
21. Muhamad Dwisnanto Putro, Duy-Linh Nguyen, and Kang-Hyun Jo, Lightweight Convolutional Neural Network for Real-Time Face Detector on CPU Supporting Interaction of Service Robot, HSI 2020, Tokyo, Japan, Jun 6, 2020.
22. Muhamad Dwisnanto Putro, Wahyono, and Kang-Hyun Jo, Multiple Layered Deep Learning Based Real-time Face Detection , ICST 2019, Yogyakarta, Indonesia, Jul 30, 2019 pp.4.
23. Muhamad Dwisnanto Putro and Kang-Hyun Jo, Real-time Multiple Face Tracking with Moving Camera for Support Service Robot, ACIIDS 2019 ACIIDS 2019, Yogyakarta, Indonesia, Apr 8, 2019.
24. Muhamad Dwisnanto Putro and Kang-Hyun Jo, Real-time Face Tracking for Human-Robot Interaction, ICT Robot 2018 2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT), Busan, Korea, Sep 6, 2018 pp.4.

Bibliography

- Akhand, M. A. H. et al. (2021). "Facial Emotion Recognition Using Transfer Learning in the Deep CNN". In: *Electronics* 10.9.
- Ban, Yuseok et al. (2014). "Face detection based on skin color likelihood". In: *Pattern Recognition* 47.4, pp. 1573–1585. ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2013.11.005>. URL: <https://www.sciencedirect.com/science/article/pii/S003132031300455X>.
- Cai, Zhaowei et al. (2016). "A unified multi-scale deep convolutional neural network for fast object detection". In: *European conference on computer vision*. Springer, pp. 354–370.
- Cao, Y. et al. (2019). "GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond". In: *Proc. IEEE Int. Conf. Comput. Vis. Works*. Pp. 1971–1980. DOI: [10.1109/ICCVW.2019.00246](https://doi.org/10.1109/ICCVW.2019.00246).
- Chen, W. et al. (2020). "STCAM: Spatial-Temporal and Channel Attention Module for Dynamic Facial Expression Recognition". In: *IEEE Tran. Affec. Comp.* Pp. 1–1. DOI: [10.1109/TAFFC.2020.3027340](https://doi.org/10.1109/TAFFC.2020.3027340).
- Chi, Cheng et al. (2019). "Selective refinement network for high performance face detection". In: *Association for the Advancement of Artificial Intelligence (AAAI)*.
- Deng, Jiankang et al. (2019). "RetinaFace: Single-stage Dense Face Localisation in the Wild". In: *arXiv preprint arXiv:1905.00641*.
- Ekman, Paul (1992). "Facial Expressions of Emotion: New Findings, New Questions". In: *Psycho. Scie.* 3.1, pp. 34–38. DOI: [10.1111/j.1467-9280.1992.tb00253.x](https://doi.org/10.1111/j.1467-9280.1992.tb00253.x).
- Elgendy, Mohamed (2020). *Deep Learning for Vision Systems*. Manning.
- Everingham, M. et al. (June 2010). "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2, pp. 303–338. URL: <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/>.

- Fu, J. et al. (2019). "Dual Attention Network for Scene Segmentation". In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* Pp. 3141–3149. DOI: [10.1109/CVPR.2019.00326](https://doi.org/10.1109/CVPR.2019.00326).
- Gan, Y. et al. (2020). "Multiple Attention Network for Facial Expression Recognition". In: *IEEE Access* 8, pp. 7383–7393. DOI: [10.1109/ACCESS.2020.2963913](https://doi.org/10.1109/ACCESS.2020.2963913).
- Hayale, Wassan, Pooran Singh Negi, and Mohammad Mahoor (2021). "Deep Siamese Neural Networks for Facial Expression Recognition in the Wild". In: *IEEE Tran. Affec. Comp.* Early Access. DOI: [10.1109/TAFFC.2021.3077248](https://doi.org/10.1109/TAFFC.2021.3077248).
- He, K. et al. (2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- He, Y. and S. Chen (2020). "Person-Independent Facial Expression Recognition Based on Improved Local Binary Pattern and Higher-Order Singular Value Decomposition". In: *IEEE Access* 8, pp. 190184–190193. DOI: [10.1109/ACCESS.2020.3032406](https://doi.org/10.1109/ACCESS.2020.3032406).
- Hossain, M. S., M. Al-Hammadi, and G. Muhammad (2019). "Automatic Fruit Classification Using Deep Learning for Industrial Applications". In: *IEEE Trans. Ind. Informat.* 15.2, pp. 1027–1034. DOI: [10.1109/TII.2018.2875149](https://doi.org/10.1109/TII.2018.2875149).
- Howard, Andrew G. et al. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arxiv:1704.04861.
- Hu, J., L. Shen, and G. Sun (2018). "Squeeze-and-Excitation Networks". In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* Pp. 7132–7141. DOI: [10.1109/CVPR.2018.00745](https://doi.org/10.1109/CVPR.2018.00745).
- Hu, M. et al. (2019). "Facial Expression Recognition Based on Fusion Features of Center-Symmetric Local Signal Magnitude Pattern". In: *IEEE Access* 7, pp. 118435–118445. DOI: [10.1109/ACCESS.2019.2936976](https://doi.org/10.1109/ACCESS.2019.2936976).
- Hu, P. and D. Ramanan (2017). "Finding Tiny Faces". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1522–1530. DOI: [10.1109/CVPR.2017.166](https://doi.org/10.1109/CVPR.2017.166).
- Jain, Vidit and Erik Learned-Miller (2010). *FDDB: A Benchmark for Face Detection in Unconstrained Settings*. Tech. rep. UM-CS-2010-009. University of Massachusetts, Amherst. URL: <http://vis-www.cs.umass.edu/fddb/index.html>.
- Jeong, Kyungjoong, Jaesik Choi, and Gil-Jin Jang (2015). "Semi-Local Structure Patterns for Robust Face Detection". In: *IEEE Signal Processing Letters* 22.9, pp. 1400–1403. DOI: [10.1109/LSP.2014.2372762](https://doi.org/10.1109/LSP.2014.2372762).

- Jiang, P. et al. (2019). "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks". In: *IEEE Access* 7, pp. 59069–59080. DOI: [10.1109/ACCESS.2019.2914929](https://doi.org/10.1109/ACCESS.2019.2914929).
- Jin, Zhong et al. (2007). "Face detection using template matching and skin-color information". In: *Neurocomputing* 70.4. Advanced Neurocomputing Theory and Methodology, pp. 794–800. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2006.10.043>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231206002840>.
- Kang, Seokhoon, Byoungjo Choi, and Donghw Jo (2016). "Faces detection method based on skin color modeling". In: *Journal of Systems Architecture* 64. Real-Time Signal Processing in Embedded Systems, pp. 100–109. ISSN: 1383-7621. DOI: <https://doi.org/10.1016/j.sysarc.2015.11.009>. URL: <https://www.sciencedirect.com/science/article/pii/S1383762115001472>.
- Kim, J. et al. (2019). "Efficient Facial Expression Recognition Algorithm Based on Hierarchical Deep Neural Network Structure". In: *IEEE Access* 7, pp. 41273–41285. DOI: [10.1109/ACCESS.2019.2907327](https://doi.org/10.1109/ACCESS.2019.2907327).
- Kim, Ji-Hae et al. (2019). "Efficient Facial Expression Recognition Algorithm Based on Hierarchical Deep Neural Network Structure". In: *IEEE Access* 7, pp. 41273–41285. DOI: [10.1109/ACCESS.2019.2907327](https://doi.org/10.1109/ACCESS.2019.2907327).
- Kumari, Jyoti, R. Rajesh, and K.M. Pooja (2015). "Facial Expression Recognition: A Survey". In: *Procedia Computer Science* 58. Second International Symposium on Computer Vision and the Internet (VisionNet'15), pp. 486–491. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2015.08.011>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050915021225>.
- Kurnianggoro, Laksono (2019). "High Performance Face Identification System with Optimized Deep Learning Architectures". PhD thesis.
- Lee, Y. et al. (2017). "Wide-residual-inception networks for real-time object detection". In: *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pp. 758–764. DOI: [10.1109/IVS.2017.7995808](https://doi.org/10.1109/IVS.2017.7995808).
- Li, Jian et al. (2019). "DSFD: Dual Shot Face Detector". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Li, Jing et al. (2020). "Attention mechanism-based CNN for facial expression recognition". In: *Neurocomputing* 411, pp. 340–350. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2020.06.014>.

- Li, Shan and Weihong Deng (2020). "Deep Facial Expression Recognition: A Survey". In: *IEEE Transactions on Affective Computing*, pp. 1–1. DOI: [10.1109/TAFFC.2020.2981446](https://doi.org/10.1109/TAFFC.2020.2981446).
- Lin, Tsung-Yi et al. (2017). "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125.
- Liu, Yang et al. (2021). "MogFace: Towards a Deeper Appreciation on Face Detection". In: *ArXiv abs/2103.11139*.
- Mathias, Markus et al. (2014). "Face detection without bells and whistles". In: *European conference on computer vision*. Springer, pp. 720–735.
- Najibi, Mahyar et al. (2017). "Ssh: Single stage headless face detector". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4875–4884.
- Nguyen, Duong Hai et al. (2019). "Facial Expression Recognition Using a Temporal Ensemble of Multi-level Convolutional Neural Networks". In: *IEEE Transactions on Affective Computing*, pp. 1–1. DOI: [10.1109/TAFFC.2019.2946540](https://doi.org/10.1109/TAFFC.2019.2946540).
- Otberdout, N. et al. (2020). "Automatic Analysis of Facial Expressions Based on Deep Covariance Trajectories". In: *IEEE Tran. Neu. Net. Lear. Sys.* 31.10, pp. 3892–3905. DOI: [10.1109/TNNLS.2019.2947244](https://doi.org/10.1109/TNNLS.2019.2947244).
- Otberdout, Naima et al. (2020). "Automatic Analysis of Facial Expressions Based on Deep Covariance Trajectories". In: *IEEE Transactions on Neural Networks and Learning Systems* 31.10, pp. 3892–3905. DOI: [10.1109/TNNLS.2019.2947244](https://doi.org/10.1109/TNNLS.2019.2947244).
- Palaniswamy, S. and Suchitra (2019). "A Robust Pose Illumination Invariant Emotion Recognition from Facial Images using Deep Learning for Human-Machine Interface". In: *Proc. Int. Conf. Comput. Sys. and Informat. Techno. for Sustain. Solut.* Vol. 4, pp. 1–6. DOI: [10.1109/CSITSS47250.2019.9031055](https://doi.org/10.1109/CSITSS47250.2019.9031055).
- Putro, M. D. and K. Jo (2018). "Real-time Face Tracking for Human-Robot Interaction". In: *Proc. Int. Conf. Inf. Commun. Technol. Robot.* Pp. 1–4.
- Putro, Muhamad Dwisnanto, Duy-Linh Nguyen, and Kang-Hyun Jo (2020). "Lightweight Convolutional Neural Network for Real-Time Face Detector on CPU Supporting Interaction of Service Robot". In: *2020 13th International Conference on Human System Interaction (HSI)*, pp. 94–99. DOI: [10.1109/HSI49210.2020.9142636](https://doi.org/10.1109/HSI49210.2020.9142636).
- Qi, DeLong et al. (2021). "YOLO5Face: Why Reinventing a Face Detector". In: *ArXiv abs/2105.12931*.

- Rawal, Niyati and Ruth Maria Stock-Homburg (2021). "Facial emotion expressions in human-robot interaction: A survey". In: *CoRR abs/2103.07169*. arXiv: 2103.07169. URL: <https://arxiv.org/abs/2103.07169>.
- Selvaraju, R. R. et al. (2017). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *Proc. IEEE Int. Conf. Comp. Vis.* Pp. 618–626. DOI: [10.1109/ICCV.2017.74](https://doi.org/10.1109/ICCV.2017.74).
- Shahbaz, A. and K. Jo (2020). "Deep Atrous Spatial Features based Supervised Foreground Detection Algorithm for Industrial Surveillance Systems". In: *IEEE Trans. Ind. Informat.* Pp. 1–1. DOI: [10.1109/TII.2020.3017078](https://doi.org/10.1109/TII.2020.3017078).
- Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.
- Sun, Wenyun, Haitao Zhao, and Zhong Jin (2018). "A visual attention based ROI detection method for facial expression recognition". In: *Neurocomputing* 296, pp. 12–22. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2018.03.034>.
- Szegedy, C. et al. (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9. DOI: [10.1109/CVPR.2015.7298594](https://doi.org/10.1109/CVPR.2015.7298594).
- Szegedy, Christian et al. (2017). "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Thirty-First AAAI Conference on Artificial Intelligence*.
- Viola, Paul, Michael Jones, et al. (2001). "Rapid object detection using a boosted cascade of simple features". In: *CVPR (1)* 1.511-518, p. 3.
- Wang, Chien-Yao et al. (2020). "CSPNet: A New Backbone that can Enhance Learning Capability of CNN". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1571–1580. DOI: [10.1109/CVPRW50498.2020.00203](https://doi.org/10.1109/CVPRW50498.2020.00203).
- Xi, Z. et al. (2021). "Facial Expression Recognition of Industrial Internet of Things by Parallel Neural Networks Combining Texture Features". In: *IEEE Trans. Ind. Informat.* 17.4, pp. 2784–2793. DOI: [10.1109/TII.2020.3007629](https://doi.org/10.1109/TII.2020.3007629).
- Yang, Bin et al. (2014). "Aggregate channel features for multi-view face detection". In: *IEEE international joint conference on biometrics*. IEEE, pp. 1–8.
- Yang, Shuo et al. (2016). "WIDER FACE: A Face Detection Benchmark". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Yang, Shuo et al. (2017). "Face Detection through Scale-Friendly Deep Convolutional Networks". In: *arxiv:1706.02863*.
- Yashunin, Dmitry, Tamir Baydasov, and Roman Vlasov (2020). "MaskFace: multi-task face and landmark detector". In: *ArXiv abs/2005.09412*.
- Zeiler, Matthew D. and Rob Fergus (2014). "Visualizing and Understanding Convolutional Networks". In: *Proc. Euro. Conf. Comp. Vis.* Ed. by David Fleet et al. Cham: Springer International Publishing, pp. 818–833.
- Zhang, K. et al. (2016). "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks". In: *IEEE Signal Processing Letters* 23.10, pp. 1499–1503. DOI: [10.1109/LSP.2016.2603342](https://doi.org/10.1109/LSP.2016.2603342).
- Zhang, Shifeng et al. (2017). "S3fd: Single shot scale-invariant face detector". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 192–201.
- Zhang, Shifeng et al. (2018). "Detecting Face with Densely Connected Face Proposal Network". In: *Neurocomputing* 284, pp. 119–127. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2018.01.012>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231218300274>.
- Zhang, Shifeng et al. (2019). "Faceboxes: A CPU real-time and accurate unconstrained face detector". In: *Neurocomputing* 364, pp. 297–309. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2019.07.064>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231219310719>.
- Zhu, Chenchen et al. (2018). "Seeing Small Faces from Robust Anchor's Perspective". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5127–5136.
- Zhu, X. and D. Ramanan (2012). "Face detection, pose estimation, and landmark localization in the wild". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2879–2886. URL: <https://www.ics.uci.edu/~xzhu/face/>.
- Zhu, Yanjia et al. (2020). "TinaFace: Strong but Simple Baseline for Face Detection". In: *ArXiv abs/2011.13183*.