NOVA IMS Information Management School

# MDSAA

Master Degree Program in

**Data Science and Advanced Analytics**

## Exploring Attention Modules for Deep Mobile Image Quality Assessment

Duarte Guerra Redinha

Master Thesis

presented as partial requirement for obtaining a Master's Degree in Data Science and Advanced Analytics

**NOVA Information Management School**
**Instituto Superior de Estatística e Gestão de Informação**
Universidade Nova de Lisboa

**NOVA Information Management School**

**Instituto Superior de Estatística e Gestão de Informação**

Universidade Nova de Lisboa

# Exploring Attention Modules for Deep Mobile Image Quality Assessment

by

Duarte Guerra Redinha

Master Thesis presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a specialization in Data Science

## Supervised by

Mauro Castelli, PhD

Illya Bakurov, PhD

November, 2023

# STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

*Duarte Guerra Redinha*

*Lisbon,25th of November of 2023*

# Acknowledgements

I want to acknowledge my deepest gratitude to Professor Illya Bakurov, and Professor Mauro Castelli for the guidance and help provided throughout the entire process of completing this work.

Professor Illya's expertise, availability, and willingness to share knowledge were instrumental in shaping the direction of this work. I am particularly grateful for the availability to discuss and guide me, even with the timezone differences.

Professor Mauro, for accepting on such short notice, to supervise my work after Professor Illya's change of location. For sharing his experience and knowledge to shape this work.

# Abstract

The use of digital media files in our society, such as images and videos, is increasing as the number of users who rely on social media applications to communicate with their peers is growing. Applications that are mostly used on mobile devices that have limited access to computational resources. The increasing importance of digital media platforms leads to the need for more suitable image quality, where the field of image quality assessment plays a vital role. The use of deep learning frameworks (convolutional neural networks) to support image quality assessment tasks has shown that it can improve state-of-the-art results. The main focus of this thesis is to improve the use of the MobileNetV3 architecture, a neural network focused on resource-constrained devices, on a full-reference image quality assessment (FR-IQA) task by changing the original attention module with three different ones. To investigate this, we propose three new MobileNetV3 models by swapping the original attention module with the Convolutional Block Attention Module (CBAM), Pyramid Squeeze Attention module (PSA), and Channel Attention Module (CAM) from the DANet network and benchmark the proposed model's performance against the original MobileNetV3 on the KADID 10K database. The experiment results showed that all three proposed models exhibited better performance on the FR-IQA task while reducing the number of parameters of the networks when compared with the original MobileNetV3. The best two variants were the model that used the PSA and the CAM attention modules.

**Keywords:** Deep Learning, Convolutional Neural Network, Image Quality Assessment, Full Reference Image Quality Assessment

**Sustainable Development Goals (SDG):**

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

Digital media platforms surround our society in every aspect of our lives. We use them to communicate with our friends and to share details and important milestones of our lives. Applications such as social media apps focus on sharing images and videos between users or with a specific group of people, and as the popularity and use of this type of application continues to grow (Auxier and Anderson, 2021), the amount of digital media files, such as images, that are shared, sent and consumed continues to increase. Image quality plays a vital role in digital communications and social media that goes beyond aesthetics. It influences the user experience by influencing choices, viewpoints, and communication. People often associate image quality with professionalism, reliability, and credibility (Li and Xie, 2019). Poor-quality images can therefore discourage interaction, reduce the impact of messages, or decrease user engagement. The need for better image quality is growing as digital media platforms become essential tools for personal and professional communication.

Image Quality Assessment (IQA) (Wang and Bovik, 2006) is a field that focuses on assessing the perceived quality of images and is divided into two classes, objective and subjective, where the latter involves direct human judgment and the former employs algorithms. There are also several tasks in IQA, depending on the availability and quality of the reference image, Full-Reference IQA (FR-IQA) when the reference image is available with the highest quality possible, Reduced-Reference IQA (RR-IQA) when the reference image is only partially available and No-Reference IQA (NR-IQA) whenever there is no access to the reference image.

The mean squared error (MSE), which was essentially simple to calculate, was the first commonly employed human-designed mathematical approach for addressing IQA problems. Nevertheless, it was unable to deliver results that correlated well with human vision perception of image quality, which can be

defined by the subjective judgment or assessment made by human observers, that represent how individuals perceive and evaluate the visual quality of an image based on their own visual experience and preferences. In light of this, the scientific community created metrics that took into account elements of the human visual system (HVS). These metrics, such as the SSIM Index (Wang et al., 2004), were based on the structural similarity paradigm and compared local similarity between the reference and the distorted images in terms of the difference in luminance and contrast between the pair and structure, rather than pixel-wise differences like mean squared error.

Deep neural networks redefined the state-of-the-art in the field of IQA, demonstrating superior performance on this perceptual task, thanks to the ongoing advancements and improvements in the Deep Learning field, more specifically in relation to computer vision (Athar and Wang, 2019). Additionally, another key aspect that has caught the attention of researchers is models of visual attention. Human attention can "focus the processing resources on the most relevant aspects of a visual scene, relegating others to the shadow of awareness" (Kanwisher and Wojciulik, 2000). Attention modules mimic this human vision behavior by enabling neural networks to focus on the most crucial features of an image while disregarding the less important. Another advantage of many attention modules is that some are lightweight and general, which allows them to be included in the majority of convolutional neural networks without significantly raising computational costs.

Including attention modules in the architecture of neural networks has been shown to improve the models' performance with only slight additional computational costs (Hu et al., 2018). Attention modules are particularly helpful in networks with limited computational resources because of their ability to improve performance without significantly increasing computational costs. One of the improvements made to create the third iteration of the mobile net architecture, MobileNetV3 (A. Howard et al., 2019), was the use of an attention module, Squeeze-And-Excite, which increased the overall performance of the MobileNet network when compared to its previous iteration, MobiletNetV2 (Sandler et al., 2018). The latest iteration has gained significant attention from both researchers and practitioners due to its suitability for mobile devices.

In recent years more attention modules have been proposed and evaluated on different neural networks and have been shown to improve the overall performance of the architectures that they were implemented on, such as the Convolutional Block Attention Module (CBAM) (Woo et al., 2018), Pyramid Squeeze Attention (PSA) (H. Zhang et al., 2023) and Channel Attention Module

(CAM) (Fu et al., 2019).

The main goal of this thesis is to investigate the possibility of improving the performance of the MobileNetV3 network on the task of FR-IQA, with the least additional computational cost increase. The big question we are aiming to answer is whether substituting the original attention module, specifically, the Squeeze-and-Excitation module, with different state-of-the-art attention modules can lead to improvements in how the network performs overall. It is also important to mention that this is a preliminary study focused on exploring the properties of MobilenetV3 architecture, in particular, we will empirically assess the suitability of the SE attention module for FR-IQA tasks. Since IQA is a relevant application in image processing regarding effective digital media management and the provision of reliable image enhancement techniques.

To evaluate the performance of the different attention modules, we decided to train and test the different modules on the KADID-10k database. This FR-IQA database is composed of 81 reference images each distorted by 25 distortions, each distortion with 5 levels of intensity. Essentially, the focus of this research is to investigate the possibility of improving the overall performance of a lightweight deep neural network on an image quality task, which is a very real practical use for this type of model. Additionally, our society is increasingly connected through social media, where images and videos are frequently downloaded, shared, and stored on cloud applications. This can be attributed as a consequence of the widespread use of smartphones and mobile applications in our society. The inclusion of this type of efficient model can introduce many advantages, for example, by assessing image quality directly on a mobile device, this type of model could reduce the data needed to store the image, compressing it to the optimal size which would result on better and more efficient storage management. Another use case is on image enhancement tools, models like the ones mentioned above can be trained on IQA problems and learn to extract hierarchical features from the training images. Then using the models as filters, it is possible to help improve image quality by increasing the brightness and contrast in low-light images for example.

The rest of this paper is structured as follows. The theoretical background is reviewed in the next chapter, where IQA and relevant developments in computer vision are discussed. The proposed approach is then discussed, including the modifications made to create the new proposed models and the consequent data manipulation required to train these models. Finally, the experimental setup is presented. In the experimental results, the results obtained from this investigation are shared. The paper concludes with the main contributions

of this investigation, together with the limitations and suggestions for future research.

# 2

# Background

## 2.1 IQA and FR-IQA

Image Quality Assessment (IQA) (Wang and Bovik, 2006) as previously mentioned in the introduction, is a crucial task in the multimedia processing field. Since the focus of this thesis is to investigate the performance of different attention modules on an IQA task, a brief discussion of the field is important. IQA focuses on bridging the gap between objective mathematical methods and the subjectiveness of human visual perception by developing metrics and models that provide results similar to the ones obtained through surveys and experiments with human test subjects.

The two main categories of IQA are subjective IQA methods and objective IQA. Subjective IQA generally involves human observers who evaluate the image's quality based on their personal visual experiences and perception of the image. Human resources are required in a controlled environment to perform subjective IQA in order to ensure consistent and trustworthy results. This need for human resources can quickly start to become increasingly costly when scaling, requiring more time to conduct and organize this type of experiment, and it can also suffer from biases based on the observers' personal experiences.

To address these limitations, objective IQA metrics were developed. These types of measures rely on algorithms and metrics that can automatically assess image quality without the need for human observers, which makes them more cost-efficient. These algorithms also have been designed to produce results like the ones obtained through the human visual system (HVS). Additionally, their results don't suffer from external biases and are easier to scale and automate without increasing their cost or needing more resources. Metrics such as the MSE and SSIM (Wang et al., 2004) are an example of objective IQA measures. Since our investigation focuses on the Full Reference Image Quality Assessment

(FR-IQA) task, we will discuss it more in-depth.

### 2.1.1 Full Reference Image Quality Assessment

When performing an IQA task and we have access to the reference image of the respective distorted image that is being evaluated, this type of situation is considered an FR-IQA problem. The objective of this type of IQA is to evaluate the degradation level in comparison to the original reference image, and at the same time, the results obtained should be correlated with the results given by the *Human Visual System*. Typical algorithms used on this type of problem are mean squared error (MSE) and mean absolute error (MAE), where the model calculates the pixel-wise average difference between the pair of images, or SSIM (Wang et al., 2004), where the structural similarity between the two images is calculated. The degree of degradation between the two images is then quantified by assigning a score based on the selected metric.

To be able to develop a deep neural network on this type of task, databases that contain reference images and the respective distorted images with different degrees of degradations and compression artifacts, along with the score of *mean opinion score* given to each distorted reference pair, are used to train the model. Databases such as TID2013 which contains 25 reference images and 3000 distorted images (Ponomarenko et al., 2015), and KADID-10K which contains 81 reference images and 10125 artificially distorted images (25 types of distortion $\times$ 5 levels of distortions) (Lin et al., 2019). The latter, KADID-10K will be the IQA Database used in this investigation because it is large and has a varied amount of reference and distortion types.

## 2.2 Deep Learning

One field of machine learning is Deep Learning, which mimics the working principles of the human brain to train models (called neural networks) consisting of neurons organized within layers. These networks are then fed with data and iteratively learn from it. At the core of the algorithm lies the neural network. It consists of interconnected layers of artificial neurons: the first layer is responsible for receiving the data as input, the second block of interconnected layers of artificial neurons is responsible for learning and understanding the patterns and features of the data, and the last block of layers generates an output for the data considered input. In essence, an artificial neural network learns

hierarchical representations of inputs and provides a prediction for a specific task (i.e., regression or classification).

These types of models have turned out to be very good at tackling problems that other fields of machine learning could not since they are highly effective at uncovering complex patterns within high-dimensional data (LeCun et al., 2015). This leads them to be very useful in many domains, such as Natural Language Processing, (Collobert et al., 2011), speech recognition (Hinton et al., 2012), and more importantly for this thesis, Computer Vision, (Simonyan and Zisserman, 2014).

One breakthrough in the Deep Learning field that was very important for Computer Vision was the development of Convolutional Neural Networks (CNN's) (Lecun et al., 1998). CNN's can, generally, be broken down into four key areas. The first one is the input layer that receives the image pixel values, then the convolutional layer where a set of filters (kernels) convolve through the input and extract potentially useful features. Next comes the pooling layer that performs down-sampling of the given input, which helps to reduce the number of parameters and increase posterior neurons' receptive field. Finally the fully connected layer performs a similar function as the output layer of a typical ANN, where scores or labels for a regression or a classification problem are calculated, respectively.

The development of the IMAGENET database (Deng et al., 2009), which at the time of its inception contained 3.2 million labeled images, and the ImageNet large scaler visual recognition challenge (ILSVRC) (Russakovsky et al., 2015), an annual object category classification competition based on the IMAGENET database, were also significant developments that enabled in the further development of these types of models. These two developments also provided a standard benchmark that could be used to assess various and new CNN architectures.

From this annual event, architectures such as VGGNet (Simonyan and Zisserman, 2014) were developed, which focused on the effect of increasing the depth of the convolutional network (number of layers), along with the use of other advancements, such as *RELU* (Krizhevsky et al., 2017). These developments allowed this network to achieve state-of-the-art results for the time being, although this performance came with disadvantages. The main one being the computational resources needed to train this networks since it had 138 million trainable parameters (the number of parameters of the VGG16 architecture). The work of (He et al., 2016) developed ResNet, which focused on increasing the depth of the network, along with the usage of shortcut connections and

residual learning blocks. With these improvements, it was shown that even though ResNet variants had up to eight times more depth than VGG, it managed to improve accuracy results in the classification task of ILSVRC by reducing the error rates from 24.4% to 21.84% on the single-model challenge. All while having fewer trainable parameters (i.e., ResNet-18 had around 11.5 million trainable parameters, which is only 8% of the parameters that VGG-16 had).

What these two models have in common is using the latest advancements in research along with developing new ones to propose networks that improve state-of-the-art performances and achieve the highest possible score in order to be the best network on ILSVRC. This leads to models that use a considerable amount of computational resources to be trained. But when we have devices with lower computation resources (i.e., smartphones), architectures like the ones mentioned previously, such as VGG and ResNet, are slow and almost unusable on those types of devices. This is where the MobileNet (A. G. Howard et al., 2017) architecture excels, developed to create an efficient and lightweight CNN that could run on resource-constrained devices. This first iteration used depthwise separable convolutions to build their neural networks. They managed to create an architecture that was nearly as accurate as VGG16 even though it was 27 times less computationally intensive. Additionally, this new architecture also had less than half of ResNet parameters, with only 4.2 million.

### 2.2.1 MobileNetV3

The MobileNetV3 (A. Howard et al., 2019) is the latest generation of the MobileNet architecture. Integrating all the advancements from previous iterations, this network improved the representational power of the model through the introduction of the Squeeze-and-Excitation (Hu et al., 2018) attention modules in the inverted residual blocks, building upon the inverted residual blocks introduced on the MobileNetV2 (Sandler et al., 2018).

In addition to these improvements two new model variants were created, MobileNetV3 Large and MobileNetV3 Small, the first one with 5.4 million parameters focused on higher accuracy performance (75.2% on IMAGENET classification task). With less than half the parameters of the larger variant, the second variant, with 2.5 million parameters, focused on optimizing resource utilization while still maintaining great performance (67.4% on the IMAGENET classification task).

Since one of the focus points of this investigation is to keep the computational

cost as low as possible, we will focus on and work with the MobileNetV3 small variant. On the table 2.1 SE represents the use of Squeeze-and-Excitation in each block, NL represents the type of nonlinearity used, where HS stands for h-swish and RE for ReLU. Additionally, s denotes stride and NBN means no batch normalization.

Table 2.1: MobileNetV3 Small Original Architecture. From (A. Howard et al., 2019).

| Input | Operator | exp size | Out | SE | NL | s |
|---|---|---|---|---|---|---|
| $224^2 \times 3$ | conv2d, $3 \times 3$ | - | 16 | - | HS | 2 |
| $112^2 \times 16$ | bneck, $3 \times 3$ | 16 | 16 | ✓ | RE | 2 |
| $56^2 \times 16$ | bneck, $3 \times 3$ | 72 | 24 | - | RE | 2 |
| $28^2 \times 24$ | bneck, $3 \times 3$ | 88 | 24 | - | RE | 1 |
| $28^2 \times 24$ | bneck, $5 \times 5$ | 96 | 40 | ✓ | HS | 2 |
| $14^2 \times 40$ | bneck, $5 \times 5$ | 240 | 40 | ✓ | HS | 1 |
| $14^2 \times 40$ | bneck, $5 \times 5$ | 240 | 40 | ✓ | HS | 1 |
| $14^2 \times 40$ | bneck, $5 \times 5$ | 120 | 48 | ✓ | HS | 1 |
| $14^2 \times 48$ | bneck, $5 \times 5$ | 144 | 48 | ✓ | HS | 1 |
| $14^2 \times 48$ | bneck, $5 \times 5$ | 288 | 96 | ✓ | HS | 2 |
| $7^2 \times 96$ | bneck, $5 \times 5$ | 576 | 96 | ✓ | HS | 1 |
| $7^2 \times 96$ | bneck, $5 \times 5$ | 576 | 96 | ✓ | HS | 1 |
| $7^2 \times 96$ | conv2d, $1 \times 1$ | - | 576 | ✓ | HS | 1 |
| $7^2 \times 576$ | pool, $7 \times 7$ | - | - | - | - | 1 |
| $7^2 \times 576$ | conv2d $1 \times 1$, NBN | - | 1024 | - | HS | 1 |
| $1^2 \times 1024$ | conv2d $1 \times 1$, NBN | - | k | - | - | 1 |

### 2.2.2 Attention Modules

Attention modules or attention mechanisms can be seen as methods to prioritize the allocation of networks' resources, directing them toward the most crucial features of the data it is handling (Hu et al., 2018). In other words, attention modules enable models to adaptively recalibrate and emphasize the most important features while suppressing less relevant information within the input data. Additionally, attention modules are usually designed to be capable of generalizing across different datasets and also to seamlessly integrate into CNN's architectures. This can be seen in the work of (Woo et al., 2018), and (H. Zhang et al., 2023) for example, making them versatile.

#### 2.2.2.1 Squeeze-and-Excitation

Squeeze-and-Excitation (SE) (Hu et al., 2018) blocks were introduced in 2017 for ILSVRC. This attention module is composed mainly by two components, the first one being the Squeeze (Global Information Embedding) Phase, which squeezes global spatial information from the entire feature map into a channel descriptor, by applying Global Average Pooling on the feature map. Generating an average value for each feature channel across the spatial dimensions. The result can be interpreted as a collection of local descriptors that represent the entire feature map. The second phase, Excitation (Adaptive Recalibration), captures nonmutual-exclusive channel relationships by using two fully connected layers, a ReLU, and a sigmoid function, and then expanding the channel dimensions, to better generalize. During this phase, channel-wise scaling factors are learned and then applied to the original feature map, recalibrating each channel's contribution.

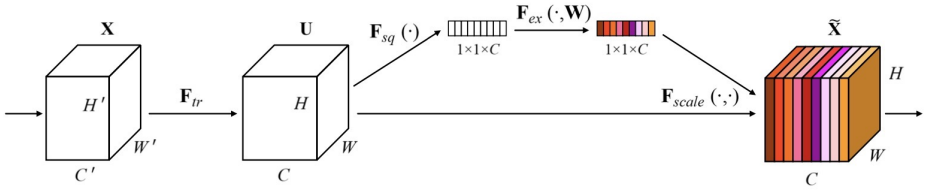


Figure 2.1: SE attention module configuration. From (Hu et al., 2018 ).

By levering these two phases, the SE block increases the model's ability to adapt to different datasets and computer vision tasks while only increasing minimally the computational costs. All while managing to integrate into different CNN's architecture as seen by the implementation on the work (Hu et al., 2018), and also (A. Howard et al., 2019).

SE block is the attention module used on the original MobileNetV3 architecture, and in our investigation, it will be used to calculate the baseline values to then compare to the other attention modules.

#### 2.2.2.2 Convolutional Block Attention Module

Convolutional Block Attention Module (CBAM) was introduced in 2018 by (Woo et al., 2018). CBAM consists of two attention modules, the Channel Attention Module (CAM) and the Spatial Attention Module (SAM). It works by receiving a feature map, sequentially inferring attention along both attention modules,

and then multiplying the attention maps with the input features map resulting in adaptive feature refinement.

The CAM emphasizes feature channels to enhance crucial channels while suppressing less informative ones. It works by generating two different spatial context descriptors from a feature map using *average-pooling* and *max-pooling* operations. After that, both descriptors are forwarded into a shared network to produce the channel attention map.

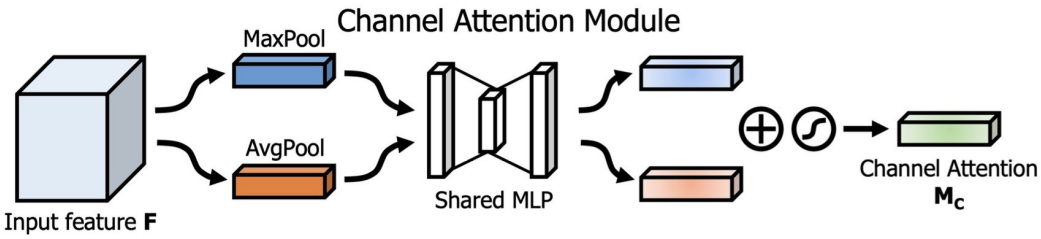


Figure 2.2: CBAM CAM module. From (Woo et al., 2018).

The second module, SAM, operates on spatial dimensions, using the inter-spatial relationship of features to generate a spatial attention map, focusing on the informative spatial locations. It initially performs *average-pooling* and *max-pooling* along the channel axis and concatenates them to produce an efficient feature descriptor. It then applies a convolutional layer on the concatenated feature descriptor to create a spatial attention map, that represents where to emphasize or suppress.

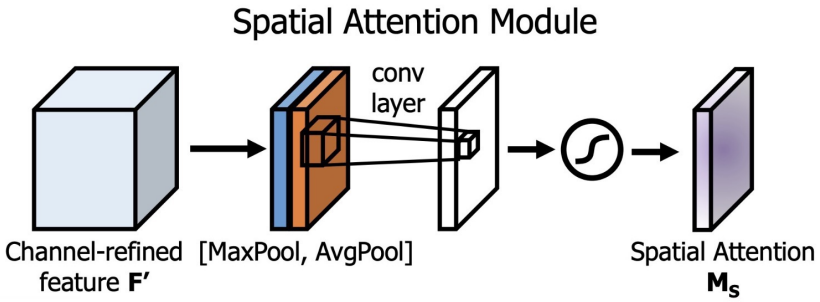


Figure 2.3: CBAM SAM module. From (Woo et al., 2018).

The lightweight and general CBAM architecture allows seamless integration into any CNN architecture with minimal computational overhead.

#### 2.2.2.3 Channel Attention Module

This attention module was introduced in the Dual Attention Network paper (Fu et al., 2019) and it is part of a network that employs the use of two attention modules together. We decided to only investigate the Channel Attention Module (CAM) since the Spatial Attention Module is too computationally heavy and would represent a drastic increase in the number of parameters and FLOPs. We decided to only investigate the Channel Attention Module (CAM) since the Spatial Attention Module is too computationally heavy and would represent a drastic increase in the number of parameters and FLOPs.

The CAM focuses on capturing inter-dependencies between channels, highlights interdependent feature maps, and enhances the feature representation of specific semantics by calculating the channel attention map directly from the original features.

It starts by reshaping the original features and performing a matrix multiplication between the original feature matrix and its transposed matrix. The result is passed through a Softmax layer to generate the channel attention map. Another matrix multiplication is performed between the original feature matrix and the transposed channel attention map, and the result is reshaped to the original feature matrix shape. The outcome is multiplied by a parameter $\beta$ (this parameter starts with weight equal to zero and gradually learns while the model is being trained) and finally, an element-wise sum operation with the original feature matrix generates the output of CAM.

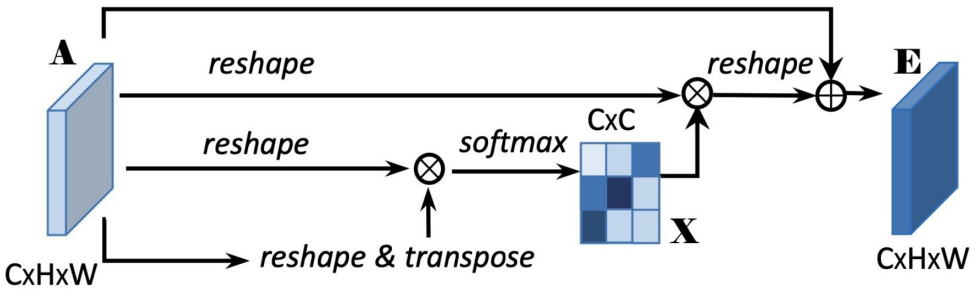


Figure 2.4: CAM attention module from DANet. From (Fu et al., 2019).

In terms of computational complexity, the CAM module is the least complex attention module among all the ones investigated in the present research.

#### 2.2.2.4 Pyramid Squeeze Attention

The most recent attention module out of the four chosen for this investigation is the Pyramid Squeeze Attention (PSA), adapted from the Pyramidal Convolution paper (Duta et al., 2020). It was utilized in 2021 on the work of (H. Zhang et al., 2023) and is comprised of four steps. It first passes the input data through the Squeeze-and-Concat module to get the multi-scale feature map channel-wise, subsequently, it uses the SE module to extract the attention of the feature maps with the different scales, obtaining the channel-wise attention vector. In the third step, Softmax is used to recalibrate and generate the recalibrated channelwise attention vector and in the fourth step, element-wise product operation is applied to the recalibrated attention weights and the corresponding feature map. This process outputs a refined feature map that contains multi-scale feature information.

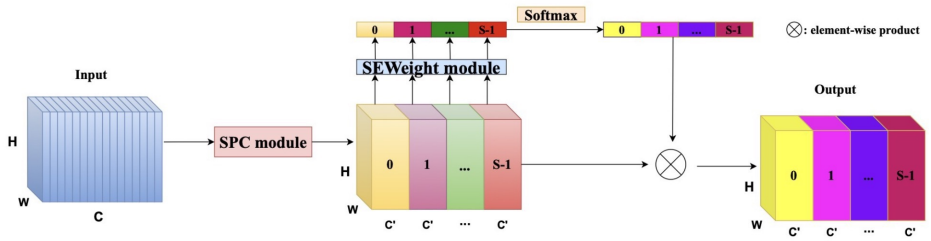


Figure 2.5: PSA attention module. From (H. Zhang et al., 2023).

The *Squeeze-and-Concat* (SPC) module is where the name of the attention mechanism is derived from, since in this module, the input is passed through a multi-way branch. In other words, the input passes through convolutions with different configurations (*kernel size, group size*) to capture more abundant spatial information of the input at multiple scales in a "parallel" fashion, and then the results are concatenated back.

It should be mentioned that this attention module, while still being lightweight, out of the four present in this thesis, is the one that adds the most computational cost to the model, mainly due to the SPC module.

## 2.3 Deep Learning in IQA

The field of IQA has undergone significant development thanks to deep learning techniques. Traditionally, engineered algorithms have served as the basis

for FR-IQA, where an image is compared against a perfect reference image. However, the development of deep learning architectures, in particular CNNs, has changed the current landscape for FR-IQA. The work of researchers using CNNs for FR-IQA tasks set the stage for understanding how these models can learn complex features and information, capture perceptual subtleties, and produce results that correlate with the human visual system.

The work of (Amirshahi et al., 2016) employed the use of a pre-trained AlexNet network to extract the feature map of the reference and the distorted images and then compare their feature similarities at different layers, using a pyramid of features' histograms similar to the Pyramid of Histograms of Orientations Gradients. The authors then evaluated and compared the performance of their proposed approach on FR-IQA databases, such as the TID2013, against 10 traditional IQA metrics, where the CNN approach outperformed the state-of-the-art metrics in most datasets.

The work of (Gao et al., 2017) proposed a framework called DeepSim. It used the traditional FR-IQA metric, SS-SSIM, to compare the reference-distortion feature maps extracted from a pre-trained VGG16, and the local quality indices at each layer were then gradually pooled to obtain a single quality score. They evaluated its performance across four FR-IQA databases, Categorical Subjective Image Quality (CSIQ), Laboratory of Image and Video Evaluation (LIVE), LIVE multiply distorted (LIVEMD), and Tampere Image Database 2013 (TID2013). The outcome of their investigation showed that CNNs are capable of undertaking FR-IQA tasks by learning discriminative features from large datasets. The DeepSim framework outperformed previous state-of-the-art FR-IQA techniques across all four databases.

Similarly, (Bosse et al., 2018) also introduced a deep neural network-based approach to FR-IQA, using the VGG as the basis for the proposed network. The architecture is compromised by 10 convolutional layers, 5 pooling layers, and 2 fully connected layers. To address the FR-IQA task in an end-to-end fashion, they used a so-called siamese network setup. The authors performed an artificial data augmentation by randomly sampling patches of images and assigning the quality labels from the original images. They also decided not to normalize the patches to allow the system to deal with the distortions introduced by luminance and contrast. The results obtained by the proposed framework outperformed state-of-the-art FR-IQA methods, such as SSIM or FSIM, and also demonstrated a great capacity for generalization across different types of distortions.

The authors of (R. Zhang et al., 2018) introduced a new FR-IQA dataset by employing traditional and CNN-based distortion types and then evaluated

deep features across three different CNN architectures, VGG, AlexNet, and SqueezeNet, and compared them with the classic FR-IQA metrics. Their findings showed that deep features outperform handcrafted metrics by large margins and that the result is consistent across all the deep architectures and levels of supervision considered in their investigation.

The work of (Varga, 2020) proposed a framework based on a Siamese layout, where pairs of images (reference and distortion images) are run through a pre-trained ImageNet database CNN architecture with shared weights to extract the feature maps. Those feature maps are transformed into feature vectors using different global pooling techniques. Additionally, a global feature vector that characterizes the reference and distorted image pair is compiled and passed to a neural network consisting of four fully connected layers used to predict the perceived image quality between the pair. The proposed framework also handled input images without the need for resizing transformations, unlike most of the previous methods. This allowed the network to effectively learn quality-related features of the whole image.

Their research focused on investigating which pre-trained CNN architecture performed the best as the body for the Siamise network, where models like the VGG-16, ResNet-101, and Inception-V3 were trained and evaluated using the KADID 10K dataset. The proposed network, called DualCNN, ended up using the Inception-V3. Finally, they compared the proposed model against state-of-the-art FR-IQA measures, such as SSIM, MS-SSIM, and DeepQA, a deep learning IQA measure (Kim and Lee, 2017). The results obtained indicated that on metrics such as PLCC, SRCC, and KRCC, the DualCNN network outperformed traditional and deep learning-based state-of-the-art algorithms on the KADID 10K database.

The combination of FR-IQA and Deep Learning methods, specifically CNNs, has greatly improved the state of the art. These models' capacity to mimic human vision highlights their promise as reliable and effective instruments for objective assessment of image quality in a variety of contexts.

# 3

# Proposed Approach

In this investigation, we aim to introduce a possible improvement to the MobileNetV3 architecture by replacing the native Squeeze-and-Excitation (SE) attention module with three alternative attention modules the Convolutional Block Attention Model (CBAM), the Pyramid Squeeze Attention (PSA), and the Channel Attention Module (CAM). The goal of this modification is to optimize the architecture's performance in Full Reference Image Quality Assessment (FR-IQA) tasks. First and foremost, we aim to train these novel architectures on a specialized task, which involves conducting pairwise comparisons between distorted images and their corresponding reference images. The model then attempts to identify and assess the difference in perceived quality between each image pair by providing a score.

We have opted to use the large KADID 10K image database in order to properly evaluate the effectiveness of these modifications. This database is comprised of 10,125 distorted images and their respective reference images. The KADID 10K database contains a wide variety of distortions, that provide a solid foundation for training and assessing our configured model variants. Through the use of this large-scale dataset, we aim to determine the model's ability to detect and measure perceived quality differences between different image pairs, highlighting the effectiveness of the suggested attention module replacements in the MobileNetV3 architecture.

## 3.1 Creating the Model Variants

To start, we use the mobilenetv3_small model as the base architecture, more precisely the *PyTorch* implementation of this model. Four new model variants are created on this investigation, mobilenetv3_base, mobilenetv3_cbam, mobilenetv3_psa and mobilenetv3_cam.

### 3.1.1 Transformation for all Variants

The source code of the original mobilenetv3_small model required some changes before introducing the 3 new attention models proposed since the model was originally designed for a classification task in the IMAGENET database. In the case of this research, the different models proposed will have to solve an FR-IQA task, which is a regression task where the models will have to return a continuous result between 1-5.

The fact that the original model was designed with a classification task in mind means that it is necessary to replace the classification head with a regression head so that the model can provide the correct results.

This new regression head will have specificities that come from the characteristics of an FR-IQA task, more specifically, a pairwise comparison between the reference image and its distorted image. Due to this type of task, pairwise comparison, the format of the input data will be quite specific. The input data is a batch of $n$ images, where the first $\frac{n}{2}$ images are distorted images and the remaining $\frac{n}{2}$ images are the respective reference images of the first $\frac{n}{2}$ distorted images. Knowing that this is the data format and that we want this new regression head to be able to provide a quality score of the distorted image in relation to its reference image, the regression head needs to be able to capture the difference between the two images. The format of the input data will have a big influence on the design of this new regression head.

In order to be able to capture the difference between the distorted images and the reference images, after the batch of images has passed through all the convolution layers and we have the feature map, it is divided into two in the batch dimension, where the first half corresponds to the distorted images feature map (fmap_dist) and the second half corresponds to the reference images feature map (fmap_ref). We then calculate the difference between the distorted feature map and the reference feature map, which represents the differences between the distorted image and the reference image, and the consequent result is fmap_diff. The feature map that represents the difference between each pair of distorted and reference images (fmap_diff) is then passed through an Average Pooling Layer to help the model reduce the chances of overfitting and reduce complexity. After the Average Pooling Layer, it passes through a one-unit dense layer with ReLU to reduce the output dimension to the final score provided by the model. This regression head schema implementation can be seen in figure 3.1.

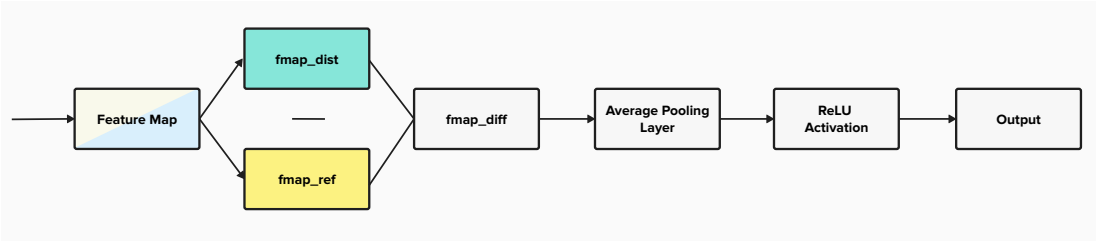


Figure 3.1: New regression head schema.

### 3.1.2 Transformations for Each Variant

#### 3.1.2.1 Mobilenetv3_cbam

To create the mobilenetv3_small variant with the Convolutional Block Attention Module (mobilenetv3_cbam) two changes to the original model were made. The first was to change the classification head to the regression head mentioned in the section above, so the model could interpret the data and calculate the results. The second change involved removing the original attention model, Squeeze-and-Excitation, and replacing it with the CBAM module. The design of this attention module does not require any more changes to the mobilenetv3_small architecture.

#### 3.1.2.2 Mobilenetv3_psa

When implementing mobilenetv3_psa, it was necessary to make some changes not only to the architecture of the original model but also to the PSA attention module.

Regarding the changes to the PSA, we decided to change the convolutional kernel size of each convolution from $[3, 5, 7, 9]$ to $[1, 3, 5, 7]$ respectively. This first change comes from the fact that without it, the resulting output channels of the four convolutions on the SPC module did not match, and this is a requirement needed for the PSA module. The second change in the PSA module was to remove the groups in the first convolution and in the remaining three use the value of the output channels of each layer divided by 4 as the value for the groups. This is because for each convolution, the number of output channels must be divisible by the number of groups, and mobilenetv3_small was not designed with this limitation, which would mean that the original PSA module could not be compatible with mobilenetv3_small without this change.

It was also necessary to not use PSA in the first layer of mobilenetv3_small, as the number of channels of the first layer was too low for PSA, because the

attention module reduces the number of channels in two stages, first in the SPC module and then in SE module, where the number of channels in the first layer did not allow this reduction.

After all these changes, the original attention module of the mobilenetv3_small network can be swapped with the modified PSA module, and the classification head is then swapped with the regression head.

#### 3.1.2.3 Mobilenetv3_cam

Only two changes are needed to create the mobilenetv3_cam variant since the CAM attention module design is versatile and there are no compatibility problems when exchanging the Squeeze-and-Excitation module with CAM. The second change is to swap the classification head for the regression head and the mobilenetv3_cam variant is ready.

#### 3.1.2.4 Mobilenetv3_base

For the mobilenetv3_base variant, the only change performed was swapping the classification head with the regression head, since we want to compare the performance of the original attention module, Squeeze-and-Excitation with the three attention modules chosen in this investigation.

For each of the respective attention modules columns on the table 3.1, the checkmark represents the use of the attention module on that block.

Table 3.1: Model Variants specification. For NL, HS, RE, NBN and s see 2.1. CBAM, PSA, and CAM represent the respective attention modules.

| Input | Operator | exp size | Out | SE | CBAM | PSA | CAM | NL | s |
|---|---|---|---|---|---|---|---|---|---|
| $224^2 \times 3$ | conv2d, $3 \times 3$ | - | 16 | - | - | - | - | HS | 2 |
| $112^2 \times 16$ | bneck, $3 \times 3$ | 16 | 16 | ✓ | ✓ | - | ✓ | RE | 2 |
| $56^2 \times 16$ | bneck, $3 \times 3$ | 72 | 24 | - | - | - | - | RE | 2 |
| $28^2 \times 24$ | bneck, $3 \times 3$ | 88 | 24 | - | - | - | - | RE | 1 |
| $28^2 \times 24$ | bneck, $5 \times 5$ | 96 | 40 | ✓ | ✓ | ✓ | ✓ | HS | 2 |
| $14^2 \times 40$ | bneck, $5 \times 5$ | 240 | 40 | ✓ | ✓ | ✓ | ✓ | HS | 1 |
| $14^2 \times 40$ | bneck, $5 \times 5$ | 240 | 40 | ✓ | ✓ | ✓ | ✓ | HS | 1 |
| $14^2 \times 40$ | bneck, $5 \times 5$ | 120 | 48 | ✓ | ✓ | ✓ | ✓ | HS | 1 |
| $14^2 \times 48$ | bneck, $5 \times 5$ | 144 | 48 | ✓ | ✓ | ✓ | ✓ | HS | 1 |
| $14^2 \times 48$ | bneck, $5 \times 5$ | 288 | 96 | ✓ | ✓ | ✓ | ✓ | HS | 2 |
| $7^2 \times 96$ | bneck, $5 \times 5$ | 576 | 96 | ✓ | ✓ | ✓ | ✓ | HS | 1 |
| $7^2 \times 96$ | bneck, $5 \times 5$ | 576 | 96 | ✓ | ✓ | ✓ | ✓ | HS | 1 |
| $7^2 \times 96$ | conv2d, $1 \times 1$ | - | 576 | ✓ | ✓ | ✓ | ✓ | HS | 1 |
| $7^2 \times 576$ | pool, $7 \times 7$ | - | 576 | - | - | - | - | - | 1 |
| $1^2 \times 576$ | conv2d $1 \times 1$, NBN | - | 1 | - | - | - | - | RE | 1 |

## 3.2 Data Manipulation

This type of FR-IQA task involves comparing the distorted images with the corresponding reference images, so the model's data input must meet this requirement.

To meet this requirement, the model's input consists of a set of $n$ images, where the first $\frac{n}{2}$ are distorted images and the remaining $\frac{n}{2}$ are the reference images of the distorted images. Another feature we had in mind when preparing the data stems from the fact that we are doing cross-validation in the first phase of the study and we need to ensure that all the models will be trained with the same set of reference images. To do this, we have divided the reference images into partitions and each partition only has distortion images relative to the reference images in its partition, thus ensuring that all the models have been trained under the same conditions.

In this research, the database used is KADID 10K due to the large number of images with different types of distortion. In order to adapt and prepare the KADID 10K to meet the above conditions, we first divided the 81 reference images into partitions and separated the distorted images into their respective partitions. Then, because we were working with Pytorch, we had to create a torch.Dataset that would retrieve the distorted images, the corresponding reference images, and the perceived quality score between the pair of images.

This torch.Dataset also applies the necessary transformations to the images to make them compatible with the input requirements of the MobileNetV3 architecture. In this case, the transformations applied to the images are normalization with mean values equal to $[0.485, 0.456, 0.406]$ and standard deviation equal to $[0.229, 0.224, 0.225]$ in the respective channels, and the images are also resized to $224 \times 224$.

Since we're using PyTorch, we also need to create a torch.Dataloader for the data that will be used as input to the model. Due to the format of the batch of images mentioned in the second paragraph of this chapter, it was also necessary to create a specific function to make the stack of the first $\frac{n}{2}$ images correspond to the distortion images and the remaining $\frac{n}{2}$ to the respective reference images.

## 3.3 Loss and Evaluation Metrics

Training neural networks, in its essence, is an optimization problem, and loss functions are used as the fitness, the value that we want to optimize in our case.

The selection of loss functions is important in our neural network training procedure because they measure differences between actual labels and model predictions. We chose the L1Loss, which is essentially identical to mean absolute error, because of its well-known resilience to outliers. This characteristic, which gives all errors the same magnitude, reinforces the model's robustness against potential outliers in the dataset. Moreover, L1Loss's interpretability and simplicity allow for a greater understanding of the errors produced by the model. Its usefulness is further improved by its computational efficiency, which enables faster training without sacrificing accuracy, and it is also commonly used for the type of FR-IQA task we are performing in our investigation, (Bosse et al., 2018), its formula is as seen here:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{3.1}$$

where $y_i$ represents the predicted value, $x_i$ is the true value, and $n$ is the sample size.

The core evaluation metrics used in our research to evaluate the effectiveness of the models are the correlation coefficients between the predicted scores produced by the models and the Mean Opinion Scores (MOS) given by human evaluators. This approach, which is based on correlation coefficients, comes from the perceptual nature of the task at hand. The main concept behind Image Quality Assessment (IQA) is the attempt to develop models that mimic human

vision. Therefore, the use of correlation metrics offers an essential benchmark for determining how well the models align with human vision. For the reasons stated above, we decided to use three correlation metrics in this investigation. The first one is the *Pearson correlation coefficient* (PLCC), which quantifies the linear relationship between both variables, it is calculated as follows:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.2}$$

where $n$ is the sample size, $x_i$, $y_i$ are the individual sample points, and $\bar{x}$ and $\bar{y}$ represent the sample mean.

The second correlation coefficient used is *Spearman's Rank* (SRCC), which is a non-parametric approach to measuring rank correlation, is very similar to the PLCC, but instead of using the data points to obtain the value, it first ranks the data points and then calculates the correlation value and its formula is:

$$r_s = \frac{cov(R(X), R(Y))}{\sigma_{R(X)}\sigma_{R(Y)}} \tag{3.3}$$

where $cov(R(X), R(Y))$ stands for the covariance of the rank variables and $\sigma_{R(X)}$ and $\sigma_{R(Y}$ represent the standard deviations of the ranked variables.

*Kendall Rank correlation coefficient* (KRCC) measures the degree of association purely based on the order of the ranked data points, since it only takes into consideration the order, it is considered more robust to outliers when compared to the previously mentioned correlation coefficients. Its formula is:

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \tag{3.4}$$

where $n_c$ represents the number of concordant pairs, $n_d$ the number of discordant pairs and $n$ is the sample size.

# 4

# Experimental Setup

This chapter provides a comprehensive review of all the fundamental components shaping our experimental framework. We go into a detailed discussion of the experimental setup parameters and techniques used in our investigation to assess the attention modules' performance in the context of FR-IQA. We will also focus on the IQA database used to benchmark and measure the performance of these new attention modules.

## 4.1 KADID 10K Database

The Konstanz Artificially Distorted Image Quality Database 10K (KADID 10K) database (Lin et al., 2019) is a large-scale IQA database for deep learning FR-IQA. It contains 81 pristine images, and each image is artificially degraded by 25 types of distortions with 5 levels of degradation intensity, which totals 10125 reference-distortion pairs. By performing a subjective IQA study, 30 degradation category ratings per image, were obtained using crowd-sourcing, to provide a reliable score to each distortion reference pair. The study consisted of showing the test subjects the reference image on the left side, and the distorted version on the right and asked to rate the distorted image in relation to the reference. In total 2209 crowd workers were involved in this study. The distortion types used to artificially distort images on this database are blurs, color-related, compression, noise-related, involving brightness changes, spatial distortions, sharpness, and contrast. Additionally, this dataset is also commonly used in works that focus on IQA (i.e., (Bosse et al., 2018) and (Zhu et al., 2020)).

## 4.2 Dataset Partition and Cross-Validation

When we are training and evaluating the performance of machine learning models, the data is usually separated into training, validation, and test partitions. The training partition is utilized to train the model, the validation partition serves as an objective assessment of the model during training, and the test partition is employed to assess performance on unseen data after the model has completed training. One advantage of using a validation set is to help prevent the model from overfitting to the training partition, which can result in the model's underperformance on unseen data. We can also obtain a better idea of the model's performance on new data by assessing the model's output on the test set.

Another method used to ensure the least biased model's metrics was *K-fold Cross Validation*, which is a resampling method that splits the train data partition into *k* folds and trains the model *k* times iteratively, where one of the *k* folds is used as validation data to and the remaining *k-1* folds are used as train data. This process is repeated until all the folds have been the validation data. Since our dataset is composed of 81 reference images, we want to ensure that the different partitions are made of unique reference-distortion pairs, meaning that each reference image is exclusive to the partition it belongs. To ensure this, the reference images were used as the reference to split the different partitions and *k-folds,* and each split only had distorted images of the specific reference image.

The first stage of the experiment, consisted of training and validating all the possible combinations of model variants and hyperparameters using *K-Fold Cross Validation* technique with *k=4*. The data partition was as follows, 25% was stored aside to be used only in the second stage (16 reference images, 2000 distortion-reference pairs), and the remaining 75% for training and validation in the first stage, corresponding to 65 reference images, 8125 distortion-reference pairs. Those 75% are divided into 4 folds, where for each iteration of the *K-Fold Cross Validation* method, 3 of the folds are for training and the remaining fold for validation.

In the second stage of the experiment, we selected the best-performing models of each of the model variants to be trained and evaluated on their performance through 50 epochs. The data partition for this stage is as follows, the same 75% that was used in the first stage as the different folds for cross-validation are concatenated together and used for training (65 reference images, 8125 distortion-reference pairs), and the remaining 25% that wasn't used on the first stage, will now be used for evaluating the performance of the model

throughout the 50 epochs, to observe the different models behavior on unseen data.

## 4.3 Experimental Parameters

Our approach involved a thorough analysis of various hyperparameter configurations in an effort to identify the best model parameters. This thorough investigation aimed to identify the optimal combination that showed the highest learning effectiveness among the wide range of potential combinations that could be examined.

The two hyperparameters that we focused on in this study were *learning rate* and *batch size*. *Learning rate* is responsible for controlling how quickly a model can adapt its weights to a specific problem, the smaller it is, the more training epochs are required to significantly update the model weights, at the risk of getting stuck. The higher the value is, the faster the model can evolve but at the cost of overfitting, so finding the optimal *learning rate* for the data and the specific model is necessary to find the balance between the risk of overfitting and the model not learning efficiently. In our experiments, all models were trained with a learning rate of 0.0001, 0.001, 0.01, and 0.1.

*Batch size* corresponds to the number of samples that will be propagated as the network input before updating the models' weights. When we increase *batch size*, the memory and the time it takes to train the model increases, since we are propagating more data. On the other hand, smaller *batch size* values produce less accurate estimations since we are passing less data every time we update the weights. In this investigation, if $n$ is *batch size*, $\frac{n}{2}$ corresponds to distorted images and the remaining $\frac{n}{2}$ are the respective reference images. The values for *batch size* used were 100 and 200.

In both stages of the experiment, meaning the cross-validation and the final stage, the number of epochs defined to train the models is 50, but in the first stage since we were testing a larger number of models, we also introduced a 5 epoch *Early Stopping* criteria. Later, in the next chapter, we will discuss further how the different combinations of these parameters affected the model's learning capabilities and produced different results.

## 4.4 Experimental Setup

All the experiments were performed using Google Colab T4 GPU, using their provided notebooks.

# 5

# Experimental Results

In this section, we will discuss the results obtained from our investigation which consisted of two stages. In the first stage of the experiment, we trained and evaluated all the model variants with all the different combinations of parameters. The objective of this section was to evaluate and select the optimal parameter combination tailored for each of the four model variants. The second stage consisted of selecting all the best-performing parameter combinations for each model's variant and training each one, to evaluate how the different attention modules affected the performance of the MobileNetV3 network.

## 5.1 First Stage - Cross-Validation of all Models Variants

In the first stage, all the models were trained with all the possible combinations of the parameters that were being tested in this investigation, batch size, and learning rate for 50 epochs with an Early Stop criteria of 5 epochs. This means that for every epoch, the validation loss is calculated and if it doesn't improve for 5 epochs, the model terminates the training loop.

Some important information for the first stage is that the loss function for all the models is L1Loss (values closer to 0 are considered better). The training loss, validation loss, and the correlation coefficients (values closer to 1 are considered better) PLCC, SRCC, and KRCC are being calculated after every epoch of the training loop. Also, the correlations and loss values seen in Table 5.1 are the results obtained from the last epoch of each fold of the cross-validation, averaged between the 4 folds.

The results from this first stage of the experiment can be seen in Table 5.1 and the best performances are highlighted in bold.

Table 5.1: First stage results for all model variants. PLCC, SRCC and KRCC stands for Pearson's, Spearman's Rank and Kendall Rank correlation coefficients. Val Loss stands for Validation Loss.

| Model | Batch Size | Learning Rate | $\overline{PLCC}$ | $\overline{SRCC}$ | $\overline{KRCC}$ | $\overline{TrainLoss}$ | $\overline{ValLoss}$ |
|---|---|---|---|---|---|---|---|
| BASE | 100 | 0.1 | 0.487 | 0.484 | 0.341 | 0.511 | 0.879 |
| BASE | 100 | 0.01 | 0.422 | 0.408 | 0.287 | 0.439 | 0.987 |
| BASE | 100 | 0.001 | — | — | — | — | — |
| BASE | 100 | 0.0001 | — | — | — | — | — |
| BASE | 200 | 0.1 | 0.403 | 0.370 | 0.259 | 0.744 | 0.908 |
| **BASE** | **200** | **0.01** | **0.513** | **0.500** | **0.346** | **0.384** | **0.870** |
| BASE | 200 | 0.001 | 0.016 | 0.081 | 0.068 | 0.967 | 2.551 |
| BASE | 200 | 0.0001 | — | — | — | — | — |
| CBAM | 100 | 0.1 | 0.298 | 0.233 | 0.161 | 0.841 | 0.902 |
| **CBAM** | **100** | **0.01** | **0.469** | **0.452** | **0.316** | **0.315** | **0.727** |
| CBAM | 100 | 0.001 | -0.001 | 0.019 | 0.013 | 1.029 | 1.261 |
| CBAM | 100 | 0.0001 | -0.048 | 0.036 | 0.0250 | 0.941 | 2.713 |
| CBAM | 200 | 0.1 | 0.424 | 0.394 | 0.274 | 0.697 | 0.884 |
| CBAM | 200 | 0.01 | 0.381 | 0.350 | 0.238 | 0.406 | 0.965 |
| CBAM | 200 | 0.001 | 0.040 | 0.111 | 0.076 | 0.969 | 2.574 |
| CBAM | 200 | 0.0001 | -0.012 | 0.067 | 0.047 | 1.039 | 2.768 |
| PSA | 100 | 0.1 | 0.487 | 0.439 | 0.312 | 0.654 | 0.834 |
| **PSA** | **100** | **0.01** | **0.570** | **0.559** | **0.398** | **0.451** | **0.847** |
| PSA | 100 | 0.001 | — | — | — | — | — |
| PSA | 100 | 0.0001 | — | — | — | — | — |
| PSA | 200 | 0.1 | 0.331 | 0.267 | 0.187 | 0.796 | 0.877 |
| PSA | 200 | 0.01 | 0.542 | 0.523 | 0.365 | 0.461 | 0.816 |
| PSA | 200 | 0.001 | -0.074 | 0.013 | 0.008 | 0.829 | 2.481 |
| PSA | 200 | 0.0001 | — | — | — | — | — |
| CAM | 100 | 0.1 | 0.073 | 0.030 | 0.020 | 0.946 | 0.952 |
| **CAM** | **100** | **0.01** | **0.547** | **0.554** | **0.395** | **0.444** | **0.847** |
| CAM | 100 | 0.001 | 0.100 | 0.123 | 0.080 | 0.554 | 1.876 |
| CAM | 100 | 0.0001 | — | — | — | — | — |
| CAM | 200 | 0.1 | 0.124 | 0.068 | 0.046 | 0.941 | 0.946 |
| CAM | 200 | 0.01 | 0.420 | 0.392 | 0.272 | 0.485 | 0.911 |
| CAM | 200 | 0.001 | — | — | — | — | — |
| CAM | 200 | 0.0001 | — | — | — | — | — |

Initially, the values for learning rate that were defined to be used in this experiment were 0.0001, 0.001, and 0.01. While training the models with learning rates of 0.0001 and 0.001, the results obtained by training the models with these values for learning rates were extremely below expectations. Because the time spent training those models was still considerably high while providing

below-expected results, the decision to introduce a new value of 0.1 for learning rate and to stop testing for 0.0001 and 0.001 was made.

In this first stage, there is a clear indication that smaller values for learning rate are not allowing the models to learn accurately, which can be seen by the train and validation loss, which is always considerably large on the variants with the smaller learning rate values (For train and validation loss, the closer to zero the better the results is). When looking at the values for learning rate, we can state that the value that provided the best results at this stage of the experiment is 0.01.

Using the same logic we can also observe that for the three variants with the new attention modules and batch size equal to 100, the performance is relatively better when compared with the same variant with the same learning rate, and batch size equal to 200. The BASE model being the only one where batch size 200 obtained better results. At this stage, the best models for each model variant were selected.

For the BASE model, looking at the results obtained from training, the best value for batch size is 200, and for learning rate, 0.01. For the CBAM variant, the best-performing parameters are learning rate equal to 0.01 and batch size equal to 100, since this parameter combination had the lowest train and validation loss and the highest correlation coefficients. Finally, for the PSA variant and the CAM variant the parameter combination that showed the best results is also batch size equal to 100 and learning rate equal to 0.01.

Another important observation that can be made at this stage of the experiment is that the best-performing variants are the PSA and the CAM. After evaluating and selecting the parameter combinations that showed proof of optimal learning in comparison with the other combinations, we passed on to the next stage of the experiment.

## 5.2 Second Stage - Comparing all Variants' Performance

The final stage of this experiment consisted of training the selected models from the first stage, this time without using *Cross Validation* and *Early Stopping*, for 50 epochs to understand and evaluate if the use of different attention modules shows any significant evidence of improving the base architecture, the result of this stage can be seen on Figures below.

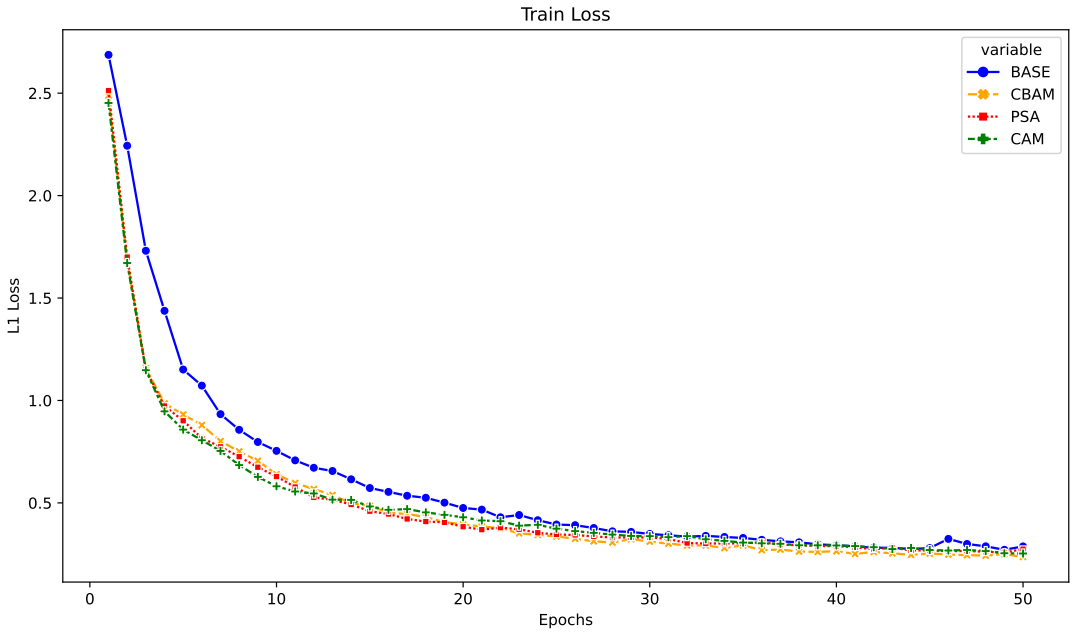


Figure 5.1: Train Loss Comparison between all variants' selected models.

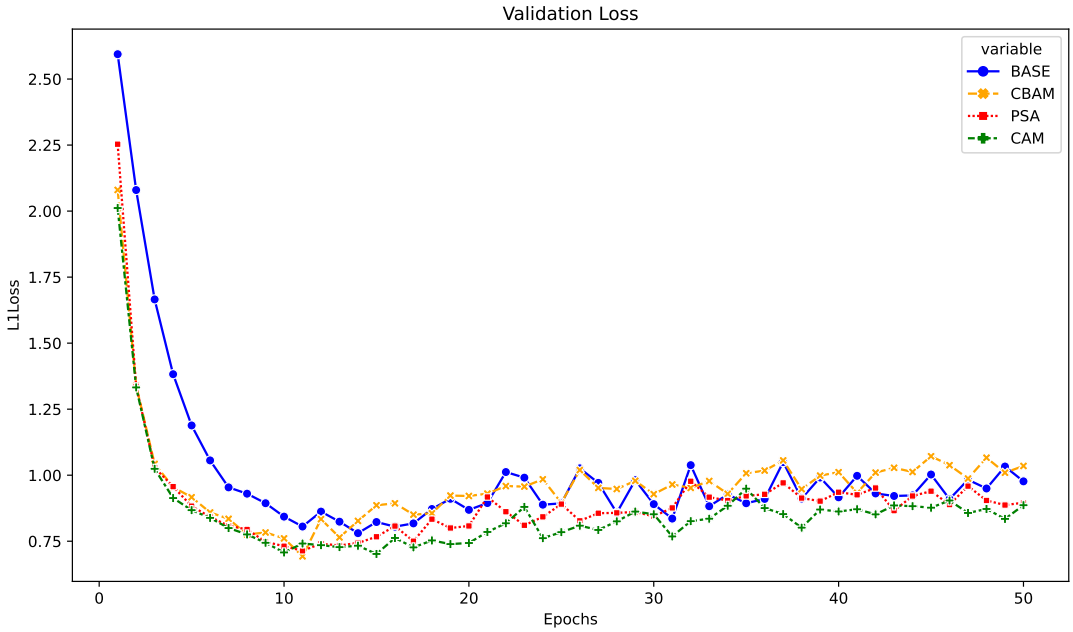


Figure 5.2: Validation Loss Comparison between all variants' selected models.

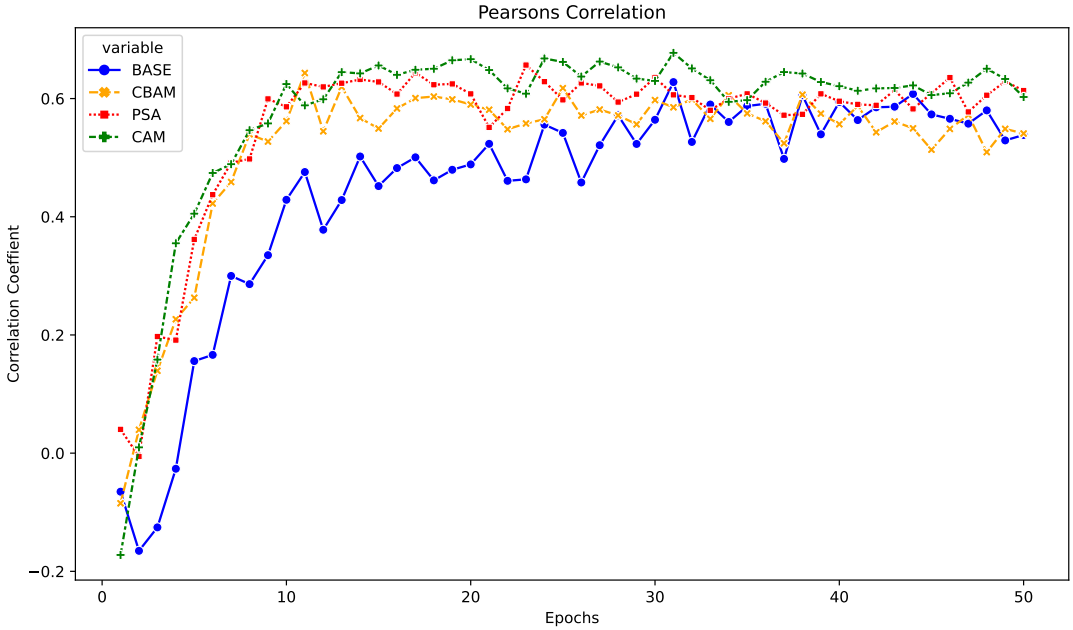


Figure 5.3: Pearson's Correlation Coefficient Comparison between all variants' selected models.

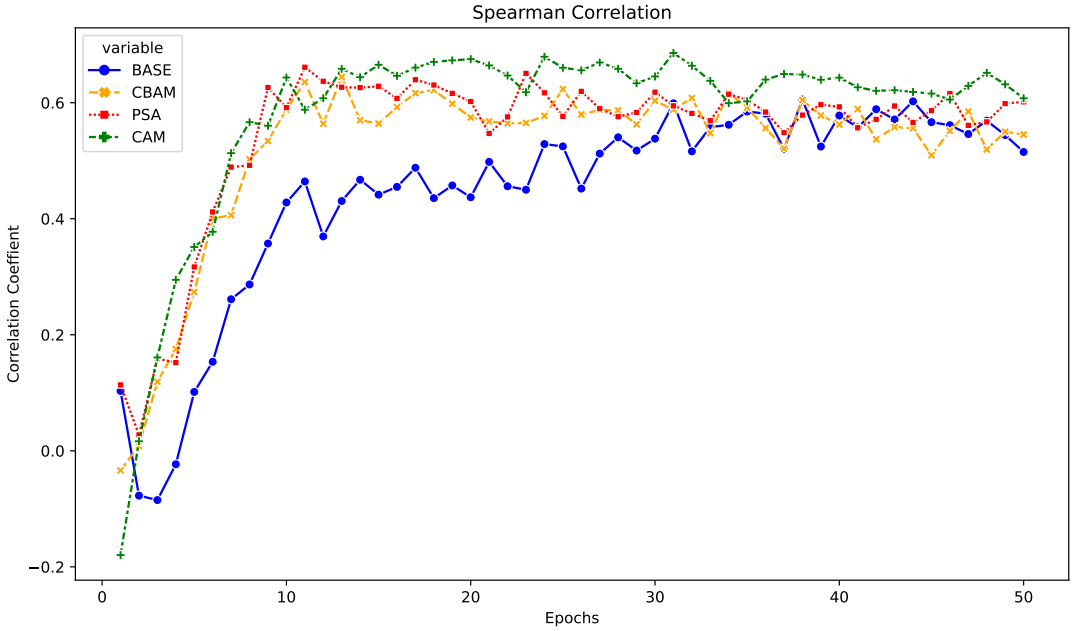


Figure 5.4: Spearman's Correlation Coefficient Comparison between all variants' selected models.

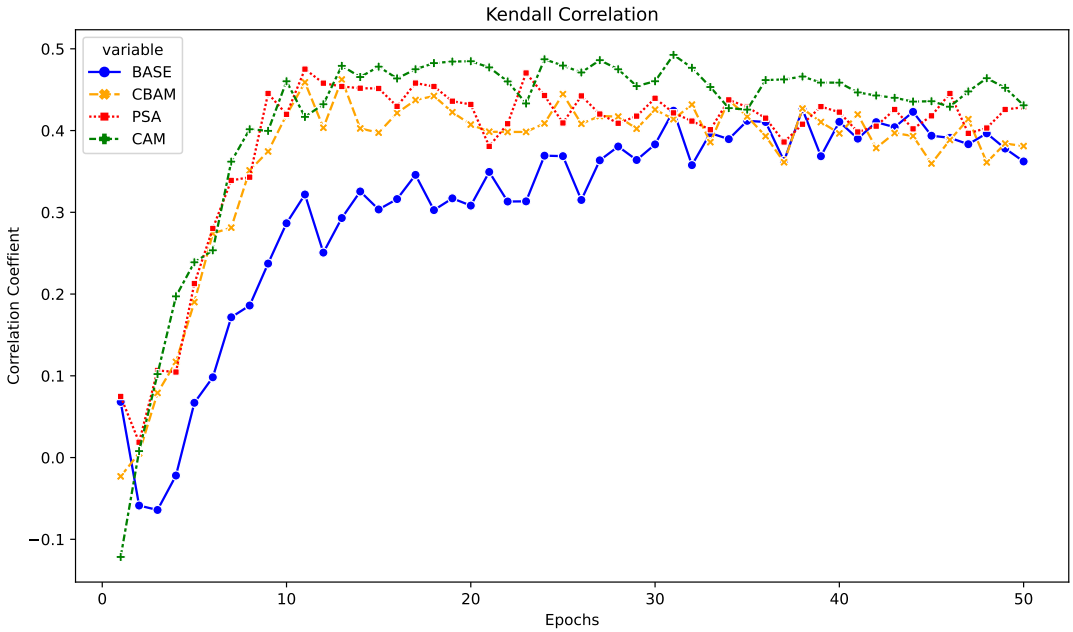


Figure 5.5: Kendall's Correlation Coefficient Comparison between all variants' selected models.

Looking at the training loss, in figure 5.1, all the models ended the 50 epochs with relatively similar values. The three variants with the new attention modules all share similar training loss values throughout the training process, but the BASE value was slower to converge onto the minimum value, displaying the worse model performance in this metric. For the validation loss values, in figure 5.2, two variants obtained the best results, the PSA and CAM, and when looking at the remaining two models, once again we can observe that the BASE model needed more epochs to obtain better results, making it the worse performing model on this metric also.

Looking at the correlation coefficient metrics, we can observe similar behavior for Pearson's correlation in figure 5.3, where the variants with BASE and CBAM obtained worse values at the end of the training process. However, when taking into consideration the whole training process, like on the previous metrics, the BASE model is slower to converge and looks more unstable when compared with the remaining models, having the worst performance on the PLCC metric. The same cannot be said about the CBAM variant. At the beginning of the training process, it evolved as fast and with similar performance when compared with the remaining two variants, CAM and PSA, just becoming more unstable in the latter epochs of the model training stage. Regarding the models with the best scores, the CAM variant was more stable when compared

with PSA.

When looking at Spearman's and Kendall's correlation graphs, figure 5.4 and figure 5.5 respectively, we can state that the trend observed in the Person's figure is also observed in both Spearman's and Kendall's coefficients. After finishing the training phase, the BASE and CBAM variants performed relatively worse when compared with the other two models, when taking into consideration the metrics from the last training epoch. Once again, the BASE model was comparatively slower to reach the same correlation values as the remaining models, more specifically, only equalling the other variants after around 30 epochs on both Spearman's and Kendall's.

When taking into consideration all the metrics, the BASE model's performance is consistently worse in comparison with all the other variants evaluated in this experiment, scoring the lowest values in all correlation coefficients (the closer to 1 the better) and highest in both train and validation loss, where values closer to 0 are considered better. Additionally, it was always the slower model to converge into the optimal value in each metric.

Table 5.2: Number of Parameters and Float Operations per Second (FLOPs).

| Variant | Params (K) | FLOPs (M) |
|---|---|---|
| BASE | 927 | 62.2 |
| CBAM | 698 | 62.1 |
| PSA | 911 | 98.2 |
| CAM | 465 | 198.8 |

Looking at the number of the parameters we can see that all the variants managed to decrease the total number of parameters when compared with the BASE model, where the CAM variant decreased the number of parameters by about 50% while performing better than the BASE model. One remark regarding the FLOPs of the CAM model, even though it is relatively high when compared with the other proposed models, it is still considered a low value when compared with other networks such as VGG-16 and ResNet. The same can be said about the remaining two proposed models, CBAM and PSA, which also decreased the total number of parameters by about 2% and 25% respectively, while also improving the MobileNetV3 network performance on an FR-IQA task.

# 6

# Conclusion

In this thesis, we investigated if the use of newer and different attention modules on a mobile deep learning network would improve the model's performance while being evaluated on an FR-IQA task, more specifically on a distortion-reference pairwise comparison. It is also important to mention that this is a preliminary investigation and the results from this can be used as a foundation for future experiments.

These types of tasks are of extreme importance for this type of model architecture since every day the number of mobile devices used around the world is increasing. Devices that are used to post, share, store, and many more features, that could benefit from the implementation of these lightweight and efficient models, for example, to enhance images, transfer them faster by compressing them without losing their perceived quality, and use less data storage when being stored, just as an example.

We proposed three new models all based around the original mobilenetv3-_small architecture by replacing the original attention model, Squeeze-and-Excitation, with three new ones, Convolutional Block Attention Module, CBAM, (Woo et al., 2018), Pyramid Squeeze Attention, PSA, from the EPSANet (H. Zhang et al., 2023) based on PyConv (Duta et al., 2020) and Channel Attention Module, CAM, from DANet (Fu et al., 2019), which were compared against the baseline, the original mobilenetv3_small. Results observed from our investigation showed that the three proposed model variants performed better than the BASE model in all the chosen metrics (Pearson's, Spearman's, and Kendall's correlation coefficients). Between the three model variants, the two models performed similarly better in general were the models with the PSA and CAM attention module, while also decreasing the total number of parameters by about 2% and 50%, respectively.

Some assumptions to provide a possible explanation for the similar behavior

shown between the two top-performing proposed variants are, for the case of the PSA variant, this attention module is the one that increases the most model complexity and has the highest number of parameters when compared with between three proposed models. It is known that increasing the model complexity leads to a higher chance of overfitting and worsened performance from the model, so this increased complexity brought by the PSA module might hinder the learning capability of the lightweight network, impeding the possibility of achieving higher results.

The CAM module was originally designed for the DANet (Fu et al., 2019) architecture, and on this network, there are two attention modules used throughout the model, the Channel Attention Module (CAM) and the Positional Attention Module, PAM, which we decided not to include on our proposed models since PAM is a resource-intensive module and added more model complexity, something that went against the objective of keeping the model as lightweight as possible. One possibility is that if both modules are used, the results observed could be superior.

Regarding the CBAM model variant, the performance observed was similar to the other two proposed methods on the starting epochs of the training process, dropping in performance on the latter epochs. While being simple to implement, the simplest out of the three in terms of complexity, and keeping the network parameter number low, these reasons could possibly be responsible for the lack of performance in the latter stages of training.

One note to have in consideration when looking at the scores and metrics achieved by the proposed models, which are below our expectations and the current state-of-the-art results, is that we had very limited access to computational resources and time to afford to run more experiments, test different parameters setups and model weights optimization to adapt the architecture to the new proposed attention mechanisms. However, we have shown that using these novel attention mechanisms showed significant improvements in the MobileNetV3 performance on FR-IQA, and with more time and resources, better scores are possible to be achieved.

# Bibliography


Amirshahi, S. A., Pedersen, M., & Yu, S. X. (2016). Image quality assessment by comparing cnn features between images. *Journal of Imaging Science and Technology*, *60*(6). https://doi.org/10.2352/j.imagingsci.technol.2016.60.6.060410

Athar, S., & Wang, Z. (2019). A comprehensive performance evaluation of image quality assessment algorithms. *IEEE Access*, *7*, 140030–140070. https://doi.org/10.1109/access.2019.2943319

Auxier, B., & Anderson, M. (2021). Social media use in 2021. *Pew Research Center*, *1*, 1–4.

Bosse, S., Maniry, D., Muller, K.-R., Wiegand, T., & Samek, W. (2018). Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Transactions on Image Processing*, *27*(1), 206–219. https://doi.org/10.1109/tip.2017.2760518

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, *12*, 2493–2537.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2009.5206848

Duta, I. C., Liu, L., Zhu, F., & Shao, L. (2020). Pyramidal convolution: Rethinking convolutional neural networks for visual recognition. *arXiv preprint arXiv:2006.11538*.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2019.00326

Gao, F., Wang, Y., Li, P., Tan, M., Yu, J., & Zhu, Y. (2017). Deepsim: Deep similarity for image quality assessment. *Neurocomputing*, *257*, 104–114. https://doi.org/10.1016/j.neucom.2017.01.054

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.90

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., & et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97. https://doi.org/10.1109/msp.2012.2205597

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., & et al. (2019). Searching for mobilenetv3. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/iccv.2019.00140

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

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2018.00745

Kanwisher, N., & Wojciulik, E. (2000). Visual attention: Insights from brain imaging. *Nature Reviews Neuroscience*, *1*(2), 91–100. https://doi.org/10.1038/35039043

Kim, J., & Lee, S. (2017). Deep learning of human visual sensitivity in image quality assessment framework. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2017.213

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Li, Y., & Xie, Y. (2019). Is a picture worth a thousand words? an empirical study of image content and social media engagement. *Journal of Marketing Research*, *57*(1), 1–19. https://doi.org/10.1177/0022243719881113

Lin, H., Hosu, V., & Saupe, D. (2019). Kadid-10k: A large-scale artificially distorted iqa database. *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. https://doi.org/10.1109/qomex.2019.8743252

Lourenço, J. M. (2021). *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. https://github.com/joaomlourenco/novathesis/raw/main/template.pdf

Ponomarenko, N., Jin, L., Ieremeiev, O., Lukin, V., Egiazarian, K., Astola, J., Vozel, B., Chehdi, K., Carli, M., Battisti, F., & et al. (2015). Image database tid2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*, *30*, 57–77. https://doi.org/10.1016/j.image.2014.10.009

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., & et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2018.00474

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Varga, D. (2020). Composition-preserving deep approach to full-reference image quality assessment. *Signal, Image and Video Processing*, *14*(6), 1265–1272. https://doi.org/10.1007/s11760-020-01664-w

Wang, Z., & Bovik, A. C. (2006). *Modern image quality assessment*. Morgan & Claypool Publishers.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612. https://doi.org/10.1109/tip.2003.819861

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. *Computer Vision – ECCV 2018*, 3–19. https://doi.org/10.1007/978-3-030-01234-2_1

Zhang, H., Zu, K., Lu, J., Zou, Y., & Meng, D. (2023). Epsanet: An efficient pyramid squeeze attention block on convolutional neural network. *Computer*

*Vision – ACCV 2022*, 541–557. https://doi.org/10.1007/978-3-031-26313-2_33

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2018.00068

Zhu, H., Li, L., Wu, J., Dong, W., & Shi, G. (2020). Metaiqa: Deep meta-learning for no-reference image quality assessment. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14131–14140. https://api.semanticscholar.org/CorpusID:215745094


**NOVA Information Management School**
**Instituto Superior de Estatística e Gestão de Informação**

Universidade Nova de Lisboa