Jônatas De Souza Nascimento

Resolution-Wise Convolutional Neural Networks for Image Classification

São Paulo, SP

2023

Jônatas De Souza Nascimento

Resolution-Wise Convolutional Neural Networks for Image Classification

Thesis submitted to the Computer Engineering Department at the Polytechnic School of the University of São Paulo.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Advisor: Prof. Dr. Artur Jordão Lima Correia

São Paulo, SP

2023

Gerar a ficha catalográfica em https://www.poli.usp.br/bibliotecas/servicos/catalogacao-na-publicacao
Salvar o pdf e incluir na monografia

To my mother

Abstract

Deep Neural Networks are the state-of-the-art solution for computer vision problems, specially image classification. However, as these architectures increase in depth and filters, the cost of these solutions also grows significantly. In this work, therefore, we study the impact of one of the parameters that plays a role on the Convolutional Neural Network's (CNN) cost: the spatial resolution of the input image. For this purpose, we develop an approach that transfers the weights of pre-trained Neural Networks to lower-scale models with the same architecture and evaluates the computational and predictive capacities of the low-scale model on the same data. Additionally, we present a method that randomly chooses the resolution for the Neural Network to be re-scaled. With the proposed methods, we achieve a reduction of Floating Points Operations (FLOPs) of 29.26% with a drop in accuracy of 2.70% using the ResNet50 on ImageNet. In particular, this was achieved reducing the resolution from 224×224 to 168×168. With the random resolution selector, we achieve a FLOPs reduction of 17.30% on the FLOPs and a negligible accuracy drop of 2.42% for the same architecture. Most importantly, these computational benefits come without any training process.

Key-words: Deep Neural Networks. Computer Vision. Image Classification.

Resumo

Redes Neurais Profundas ocupam o estado da arte em problemas de visão computacional, especialmente classificação de imagens. No entanto, à medida que estas arquiteturas aumentam em profundidade e filtros, o custo dessas soluções também cresce significativamente. Neste trabalho, portanto, estudamos o impacto de um dos parâmetros que mais influencia no custo da Rede Neural Convolucional (CNN): a resolução resolução espacial de entrada do modelo. Propomos uma abordagem que copia os pesos de Redes Neurais pré-treinadas para modelos de menor escala com a mesma arquitetura e avalia a redução de FLOPs e precisão do modelo de menor escala no mesmo conjunto de testes. Além disso, apresentamos um método que escolhe aleatoriamente a resolução para a Rede Neural ser reescalada. Com os métodos propostos, alcançamos uma redução de Operações de Ponto Flutuante (FLOPs) de 29,26% com uma queda na precisão de 2,70% na ResNet50 avaliada no ImageNet ao reduzir sua resolução original de 224×224 para 168×168. Com o seletor de resolução aleatória, alcançamos uma diminuição de 17,30% nos FLOPs e 2,42% na precisão para a mesma arquitetura.

Palavras-chave: Redes Neurais Profundas. Visão Computacional. Classificação de Imagens.

Contents

1 Introduction

Convolutional Neural Networks became the state-of-the-art approach for various tasks in computer vision. These novel developments revolutionized several fields including image classification (LI et al., 2014), sign language translation (ABIYEV; ARSLAN; IDOKO, 2020), and object detection (GALVEZ et al., 2018).

Among the efforts in improving convolutional architectures, the studies of the structures that compose these architectures stand out, namely the study of the impact of the CNNs filters and layers. Previous studies show that increasing the number of filters and layers improves the models in different metrics, including accuracy (AHMED; KARIM, 2020).

However, the increasing number of filters and layers of the novel CNN architectures (HE et al., 2016), together with the increasing resolution of the images in computer vision applications, increase significantly the cost of CNN-based solutions for these applications. This fact limits the portability of these technologies and increases the hardware requirements.

Such advancements led to the growth of the research for efficient Neural Networks (HOWARD et al., 2017; TAN; LE, 2019). Different works put effort into developing architectures that consume less computing power while preserving high accuracy, or developing pruning techniques to remove structures of the architecture that have a low impact on the predictions.

The goal of this pursuit is always to reduce the total number of parameters of the Networks. In particular, such strategies focus on consuming less computational resources, including memory and processing power, and thus, to be portable to less powerful hardware, including mobile phones.

Previous studies indicate that resolution is one of the parameters of the most direct impact on the computational resources required by a Neural Network. These developments show that bigger resolution may result in more accuracy in some contexts (HAN et al., 2020), but it also leads to more memory requirements and more floating points operations. However, there has been not much research on the possibility of reducing the resolution of models and keeping the predictive ability.

Our work, therefore, delves deeper into the resolution impact on CNNs. Specifically, we conduct experiments using popular architectures to investigate the role of resolution on the predictive accuracy of Convolutional Neural Networks.

1.1 Goal

The main goal of the present work is to study the role of resolution in modern Convolutional Neural Networks architectures. We aim to particularly comprehend the impact of reducing the resolution of pre-trained networks. Throughout this work, the exploration is made by the investigation of the following research questions:

Is it possible to reduce the resolution of a trained Neural Network model and keep its predictive capacity?

Previous results in the literature have shown that increasing the resolution improves the prediction accuracy. However, increasing this parameter also leads to more consumption of resources by the CNNs, which is often a problem.

Therefore, in this work, we study the impact of reducing the resolution of a pre-trained model. Even though this proposition can lead to interesting developments in the field of efficient Neural Networks, because of its possibility to reduce computational resources while maintaining the model's accuracy, it has not been widely investigated in the literature.

Is it possible to systematically determine a resolution for which the model keeps its predictive ability?

If the answer to the first research question is found to be affirmative, another important inquiry emerges. If it is possible to reduce the resolution of a Network and keep its predictive capacity, we propose to investigate a method to select the resolution to perform the model's predictions.

This is of high relevance because selecting a resolution lower than the one used in the training procedure of the CNN involves a tradeoff between computational efficiency and accuracy. An ideal resolution would save computing power while still providing the best accuracy.

Therefore, one of the questions that guides the research is how to define the resolution to use in the model's prediction procedure. Further questions that come up throughout the work are discussed in the next chapters.

These questions are of high significance for the research field, and it is another step towards understanding Convolutional Neural Networks. This becomes clear when considering that the results of the experiments could indicate that not all model's capacity is used to predict all samples the same way.

A possible affirmative answer to these questions could potentially contribute to the CNNs efficiency discussion. If reducing the resolution of trained models has little effect on their predictive ability, we can use it as a technique to reduce the computational costs of Neural Networks.

1.2 Methodology

The present work consisted of investigating the research questions proposed beforehand. Starting with literature research on the topic, studying and reviewing the main works that address the resolution factor on Convolutional Neural Networks and other important developments on efficient Neural Networks.

After carefully understanding the major developments on the topic, the first step was to define the scope of the experiments. We made them using publicly available datasets that are commonly used as a benchmark for image classification problems: CIFAR-10 and ImageNet.

Another important definition made was the architectures to be tested. We use architectures that are modern and achieve high performance on the two datasets. The chosen architectures were: NASNet and MobileNet for the CIFAR-10 set, and ResNet for the ImageNet set.

The experiments, described in more depth in chapter 3, consisted of using the weights from the models trained on the original resolution, to new models with a lower scale. Then these models are used to predict the classifications in the test set, and we evaluate the accuracy and the floating points operations count.

Additionally, to address the resolution selection question, we propose to choose the resolution of the model randomly and evaluate both metrics once more. This proposition is made to evaluate the gain in efficiency and loss in accuracy when choosing the resolution in a completely random fashion and is better defined in the methodology section.

The next step was to discuss the obtained results in light of the initially proposed research questions. Answers to these questions were developed based on the achieved results. Additionally, we compared these results with other propositions of computational resources' reduction in the literature and evaluated the contributions of this work.

1.3 Outline

We divide the thesis into six chapters. The present one (chapter 1) gives the reader an overview of the researched topic. It also provides a sketch of the methodology and structure of the work, together with questions that guide the research. In the following chapter (chapter 2), there is a more in-depth discussion of the state-of-art research in the field, giving more details on similar investigations, and discussing the general literature on scale-wise Neural Networks. Lastly, a deep learning theoretical background is available in the same chapter.

The chapter 3 details the methodology of the thesis. Where we define a more

in-depth explanation of the datasets and Neural Network architectures used. All of this is described together with a discussion on the approaches proposed for scale reduction and selection, the setup for the experiments, and the evaluation metrics.

As soon as the base for the experiments is well established, the final obtained results are shown in chapter 4. Then, in chapter 5, there is a deep discussion of the presented results, highlighting the findings in light of the initially proposed research questions. In this section of the work, a balance of the proposed approach is conducted, where the significance of the present work for the computer vision field is evaluated. The conclusion, in chapter 6, discusses the final aspects of the thesis by providing insights and highlighting further questions related to this project.

2 State of the art

In order to be able to investigate the proposed research questions, a deeper analysis of the literature work that studies the resolution impact on Convolutional Neural Networks is necessary. Therefore, the present chapter provides an overview of the recent developments on the topic, including other relevant studies for the thesis' discussion.

2.1 Efficient Neural Networks

The search for efficient Convolutional Neural Networks is not new, since these architectures became the state-of-art for computer vision problems, it also became clear the necessity to reduce the computational costs of their implementations.

(HOWARD et al., 2017) made an important study on the topic, with an extensive discussion on the accuracy-resource tradeoff, and proposed a CNN architecture called MobileNet, where two shrinking parameters are simply defined and used to improve the model's efficiency.

In the same direction, (ZHANG et al., 2018) proposed two different operations to drastically reduce computational costs, which they called pointwise group convolution and channel shuffle. Their experiments showed significant improvement in accuracy in comparison to MobileNets, and lower model complexity.

These results were of high importance to applying Neural Networks solutions in environments where computational resources are scarce. Mobile applications are perhaps an important example of a situation where the models need to run with a reduced amount of computing power.

In a more disruptive study, (TAN; LE, 2019) proposed a systematical method to uniformly scale depth, width, and resolution of Neural Networks, and use Neural Architecture Search (NAS) to design EfficientNets, CNNs that are highly efficient, achieving the best accuracy on TOP5 ImageNet benchmark at the time, while being 8 times smaller than the best ConvNet at the time.

Based on this study, (HAN et al., 2020) showed that resolution and depth have more impact on smaller Networks than width. Therefore, they incremented the EfficientNet proposition with a smaller formula to downsize ConvNets, and based on this formula defined the TinyNets, which have similar costs to MobileNets and much higher performance.

More recently, (VASU et al., 2023) studied the correlation between FLOPs and parameter count with latency in real-world application scenarios. Based on their results,

they addressed the bottlenecks of previous efficient networks and proposed a mobile network called MobileOne, especially focused on improving CNNs latency on mobile phones. Their final results include state-of-the-art performance while having faster networks than previous models.

2.2 Resolution

The previously discussed studies highlight the extreme impact of resolution on the Convolutional Neural Networks efficiency. Especially, (HAN et al., 2020) made experiments using FLOPs constraints and attested that the resolution is the factor that has the most impact on the accuracy of their models.

In the medical imaging field, (THAMBAWITA et al., 2021) studied the impact of changing the resolution of the CNNs and images in the classification of endoscopy images. In their experiments, different Neural Networks are trained with different resolutions, and evaluated also in different resolutions. The conclusion is such that higher resolutions lead overall to better accuracy and not much more processing time. Part of their experiments is similar to the ones we describe further in the thesis, although a comparison is not directly possible given the distance between the applications and evaluation metrics.

Perhaps one of the most complete works made on the Convolutional Networks resolutions was made by (TOUVRON et al., 2019). In their work, they investigate the discrepancy between the train and test resolution originated from data augmentation procedures.

An important discussion made by this work is that the resizing operation, applied to rescale the original images to the input shape of the Networks, changes the apparent size of objects in the image. Therefore, what the Neural Network "sees" is different from the actual image.

This has an impact when data augmentation is used in training procedures. The study shows that for augmentations commonly used, the apparent size seen in the CNN training is significantly different than the apparent size seen in the test procedure.

To address the discussed discrepancy, they propose a fine-tuning of the network, on its last layers, using the correction factor for the images to have the same apparent size that they have when testing.

2.3 Deep Learning Background

This section provides a theoretical background on the deep learning techniques used extensively in this thesis. Sections 2.3.1 and 2.3.2 give an overview of neural networks

and discuss their relevance to the computer vision field.

2.3.1 Neural Networks

Neural networks are one of the most investigated and used machine learning techniques in this day, and they have special relevance in various fields, including computer vision classification tasks, which is the main focus of this thesis.

Such models are often compared to biological neural networks, and the comparison comes from the original architecture of neural networks, which simply consist of a set of connected units that process information. The units are called *neurons* or *perceptrons*, and each of them usually processes information as described in Equation 3.2. Figure 1 is a representation of the Neural Network unit.

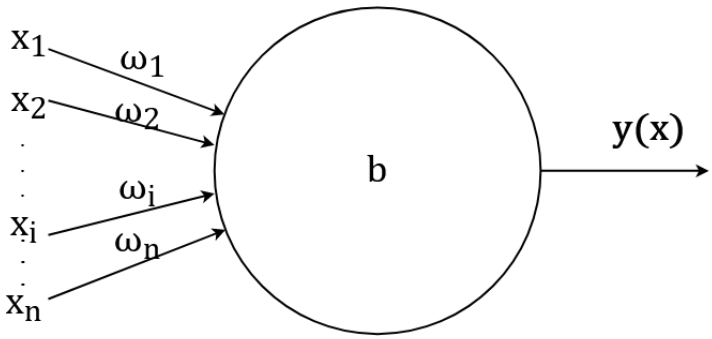

Figure 1 – Neuron representation

$$\boldsymbol{y}(\boldsymbol{x}) = h(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b). \tag{2.1}$$

In the formulation above, $\boldsymbol{y}(\boldsymbol{x})$ is the output, $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ is the input vector, $\boldsymbol{w} = (w_1, w_2, ..., w_n)$ is the weight vector, b is the bias and $h(.)$ is the activation function, which is the factor used in neural networks to provide non-linearity capabilities. Commonly used activation functions (LUNDERVOLD; LUNDERVOLD, 2019) are shown in 2.2.

Activation Function	
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$
Softmax	$\sigma(x) = \frac{e^{x_i}}{\sum_j e^{x_j}}$
Tanh	$\sigma(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
Rectified Linear Unit (ReLU)	$\sigma(x) = (0, x)$
Exponentional Linear Unit (ELU)	$\sigma(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha(e^x - 1), & \text{otherwise} \end{cases}$

(2.2)

A neural network is, therefore, simply a set of connected neurons, typically organized into layers, where its outputs are evaluated in cascade. Figure 2 provides a standard Neural Network architecture. In the figure, each neuron in the so-called hidden layer processes the results of every neuron in the layer before and has its own weights, biases, and activation function. Equation 2.3 formalizes the Neural Network output, following the referred architecture with m layers.

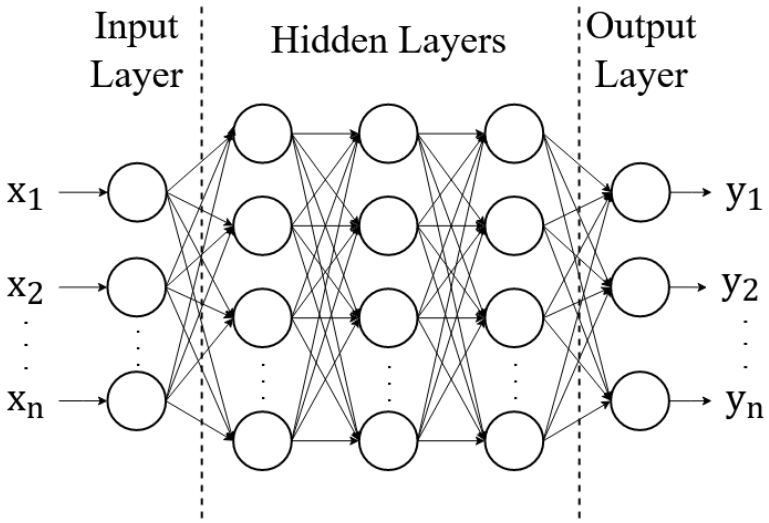

Figure 2 – Neural network representation

$$y(x;\Theta) = (y_m \circ ...f_1)(x) = y^m(h^{m-1}(...(h^1(\boldsymbol{w^T x} + b_1) + b_{m-1}) + b_m) \tag{2.3}$$

Where $\Theta = \{\boldsymbol{w_1}, ..., \boldsymbol{w_m}, b_1, ..., b_m\}$ is the trainable parameter set. The large number of parameters and the high degree of non-linearity caused by the activation functions enable the Networks to capture complex characteristics of datasets. Thus, when the parameters

are correctly found, neural networks can exhibit outstanding performance when addressing highly complex problems.

However, training a big set of parameters such as the one present could potentially be a challenge, this process is usually made based on gradient descent optimization (RUMELHART; HINTON; WILLIAMS, 1986), which is an algorithm that updates the weights iteratively, towards the vector direction where the error is lower. Equation 2.4 formalizes the iterative update of the gradient.

$$\boldsymbol{w}^{\tau+1} = \boldsymbol{w}^{\tau} - \eta \nabla E(\boldsymbol{w}^{\tau}) \tag{2.4}$$

The weight vector $\boldsymbol{w}$ is updated for the next by subtracting the factor η multiplied by the gradient $\nabla E(\boldsymbol{w}^{\tau})$ for each step τ. The factor η is the learning rate parameter and can be chosen in order to make the Neural Network learn faster or slower, while the gradient of the error gives the information towards which direction we have to update the weights in order for it to decrease.

Even though it can take much processing time to calculate the gradient if done naively, it can be simply evaluated by a technique called backpropagation (RUMELHART; HINTON; WILLIAMS, 1986), which is an algorithm that efficiently evaluates the gradient of the error function $E(\boldsymbol{w})$ of a Neural Network. Its main advantage is its computational efficiency, since it can evaluate the current error in $O(\boldsymbol{w})$ steps(GOODFELLOW; BENGIO; COURVILLE, 2016). This efficiency is central for the neural networks training process to be relatively fast, even in networks with a high number of neurons.

The powerful capacity of neural networks, aligned with the simplicity and high efficiency of the training process, made them one of the most known and applied machine learning techniques for the most diverse problems. The most diverse adaptations and modifications to these models were made over the years, and one of the most relevant for the present thesis is the Convolutional Neural Network, which is described in 2.3.2.

2.3.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a special type of Neural Network used commonly to address problems involving image data. It is the main tool used in the most variety of computer vision problems (FENG et al., 2019).

Networks following the architecture described in 2.3.1 are usually called fully connected networks, and they generally serve well for tasks with a controlled number of input parameters. However, when escalating this approach to high-resolution images, which often have over 10^4 input pixels, the number of trainable parameters increases quickly.

Fully connected networks, therefore, have a large number of connections and

weights, which cause strong limitations on training. Especially those called *deep neural networks*, which have a significantly high number of layers. For those cases, even using high-performance hardware takes a long time in training and consumes too much memory and processing power.

Convolutional Neural Networks propose a way around this problem, reducing the number of trainable parameters while still keeping the capability of training with a high number of parameters in the inputs (GOODFELLOW; BENGIO; COURVILLE, 2016). This is done via a shared parameter technique. In this case, each neuron in a CNN layer only receives input from a reduced map of the layer before, and the weights are shared for each neuron in these layers, which iterate over all sections of the previous layer to produce the next layer. This set of shared weights is called the kernel filter, and the same backpropagation technique is used to train these parameters. Figure 3 illustrates such operation.

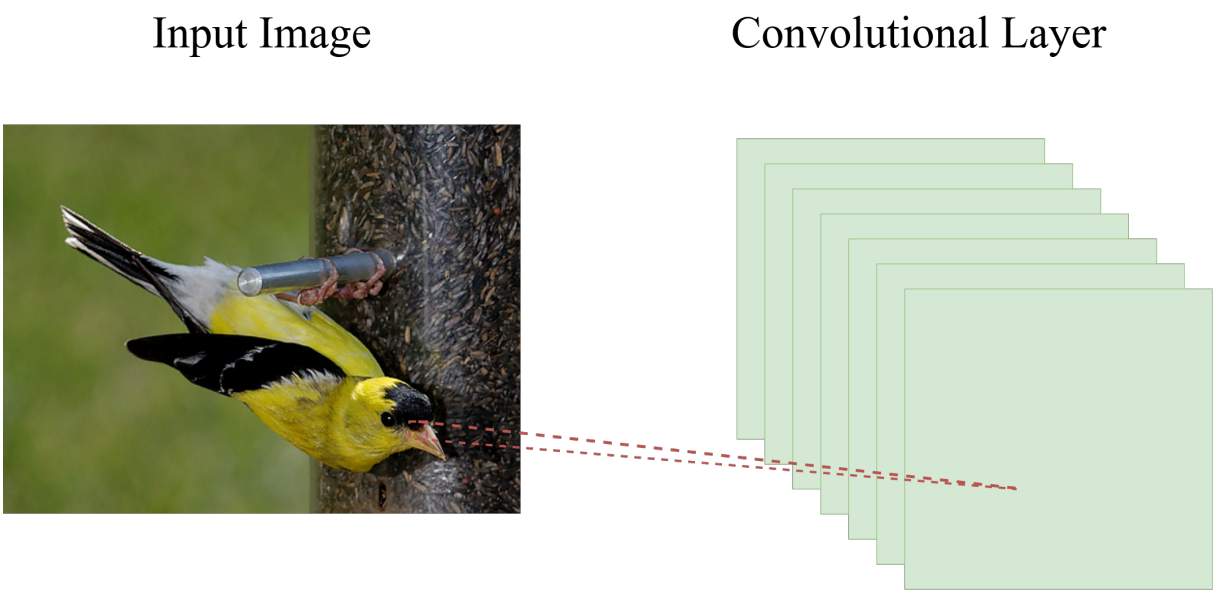

Figure 3 – Convolutional layer representation

The described operation is a convolution of the input X with a filter kernel H given a receptive field $K \times K$. Equation 2.5 formalizes the convolutional operation.

$$\boldsymbol{Y}[n,m] = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \boldsymbol{H}[i,j] \times \boldsymbol{X}[n+i, m+j] \tag{2.5}$$

Usually, more than one filter kernel is learned by layer in a convolutional network, which does not excessively inflate the parameter count. For instance, a convolutional layer with 128 kernels, each with a 3×3 receptive field, has only $128 \times 3 \times 3$ parameters even when trained in 224×224 images. In comparison, a fully connected layer for the

same input, when having the same output size, needs to train $(224 \times 224)^2 + 224 \times 224$ parameters, which is of an order of 10^6 higher.

Having multiple kernels is relevant because these structures are often responsible for learning different patterns in the image. For instance, filters can identify edges or ridges in the images. The number of convolutional layers also influences these identifications, kernels close to input often identify low-level features, whereas kernels close to input can identify object parts (ZEILER; FERGUS, 2014).

According to Figure 3, full-resolution images have a significant amount of repetitive information in different pixels. Therefore, another structure in the CNN architecture is responsible for improving its efficiency. This structure, called pooling layer, is responsible for downscaling the image and removing redundant information. Thus, the layer receives the max or average value of every 2×2 region of the image and passes the down-sampled pixels to the next section of the network. Pooling layers are often added after convolutional layers, and one 2D representation of this operation can be seen in Figure 4.

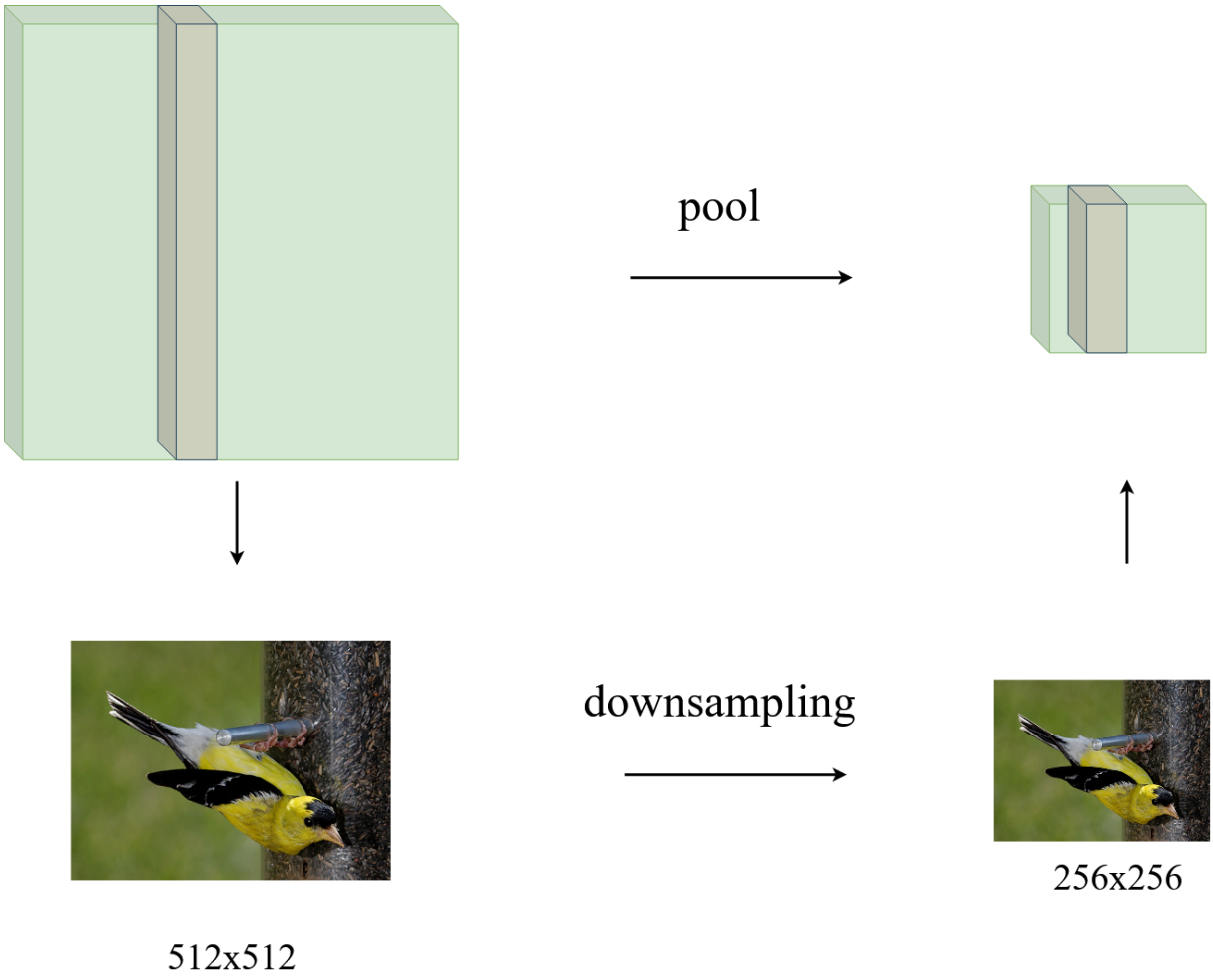

Figure 4 – Pooling layer representation

The pooling operation is of significant relevance for Convolutional Networks imple-

mentations, not just because it reduces the computational power, but also the memory required during the training procedure and to store the models. Still, literature has also shown the capability of these layers of reducing overfitting (SCHERER; MüLLER; BEHNKE, 2010).

Finally, a usual CNN architecture consists of a set of N convolutional and pooling layers connected to M fully connected layers, followed by the output. Figure 5 shows a representation of the mentioned process

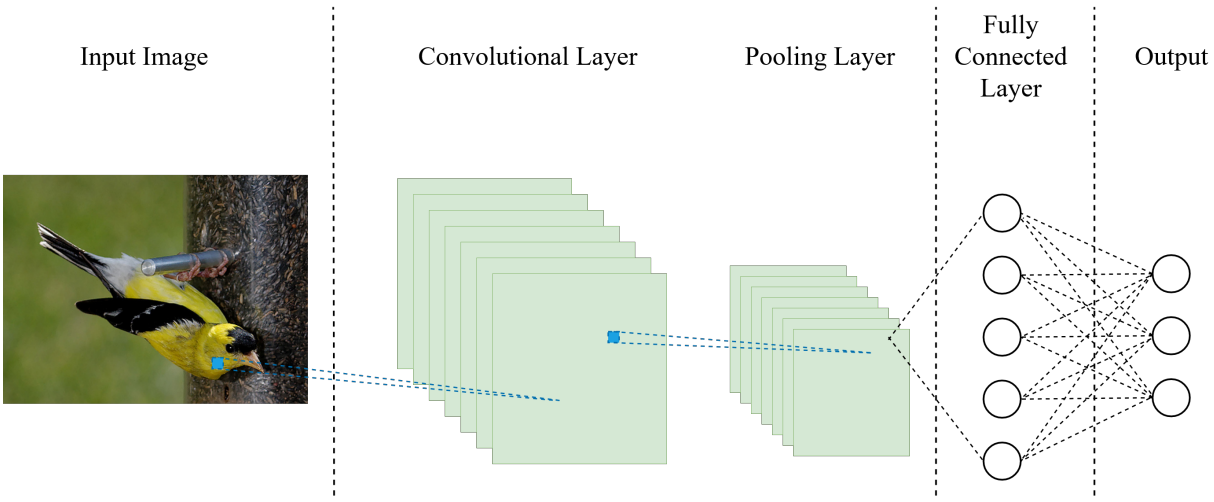

Figure 5 – CNN architecture representation

3 Methodology

Based on the research questions, and the analysis made in chapter 2, where we attested the resolution role in the accuracy of Convolutional Neural Networks, we propose to reduce the test resolution on pre-trained Networks. For this purpose, we investigate the impact on two main metrics: accuracy and Floating Points Operations (FLOPs).

This chapter contains a detailed description of the methodology conducted throughout this work, including an overview of the used datasets, the proposed method of resolution reduction, and a description of the metrics used to evaluate the models.

3.1 Dataset Description

To conduct the experiments in this work, we use two datasets. The first is CIFAR-10, where we made the experiments on lower-resolution images, and the second is ImageNet, where we conducted the experiments on higher-resolution images and comparison with existing works.

3.1.1 CIFAR-10

CIFAR10 is a dataset widely used to evaluate image classification of Deep Neural Networks. The dataset consists of 60,000 images in a relatively low scale (32×32 pixels) and contains ten classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. From the 60,000 images, 50,000 are designated for training and 10,000 for testing, each category having 1,000. This dataset was first described in (KRIZHEVSKY; HINTON et al., 2009).

An important remark about CIFAR-10 is that the quality of the images varies. The pictures are in several conditions of illumination and pose, and many objects are occluded. Therefore, even human recognition of these images is affected, and studies indicate that the evaluated accuracy for humans is around 93% (HO-PHUOC, 2018).

3.1.2 ImageNet

ImageNet is the most used dataset for image classification and object recognition tasks, being first introduced in (DENG et al., 2009). This set contains over 15 million high-resolution images, labeled in over 22000 categories. Often, the images on the set contain objects from more than one class, however, for classification problems, each image has one single label.

The images' resolutions are varied within the dataset. However, most deep learning frameworks requires a constant dimension as an input, and therefore the images are always rescaled to a fixed resolution throughout this work. This rescaling feature is present in the modern Convolutional Network architectures, given their fixed input resolution.

3.2 Qualitative metrics

This section describes the metrics used throughout the work to provide a quantitative measure of the proposed approaches, and therefore provide direct comparisons between them and existing works.

Consequently, the metrics chosen are commonly used in the literature for similar tasks. In classification problems, accuracy and TOP-K accuracy are the main ways to evaluate Convolutional Neural Network models. On the other hand, cumulative Floating Point Operations (FLOPs) are often used to compare the efficiency of the models. All these metrics are described in more detail in this section.

3.2.1 TOP-1 Accuracy

TOP-1 accuracy is one of the most important metrics in the classification tasks context, it measures the proportion of examples for which the predicted class matches the target label. For most use cases, it is simply called *accuracy*.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{3.1}$$

Even though this metric is widely used for classification tasks, and will be used as the standard metric to evaluate the predictions on the CIFAR-10 set in this thesis, there is a remarkable issue when using it for our use cases. As discussed beforehand, the ImageNet dataset has images that contain objects from more than one single label. Therefore, for this metric, the Neural Network could perform a correct prediction, and still be considered wrong.

Consequently, using TOP-1 Accuracy will undoubtedly lead to imprecisions, and not fully represent the quality of the models when evaluating them on the ImageNet dataset. Hence, another metric commonly used in the literature will be used as the standard in this thesis for this case.

3.2.2 TOP-K accuracy

To address the discussed issue with using TOP-1 accuracy on the ImageNet dataset, we use TOP-5 accuracy to quantify the results for the ImageNet experiments. This measure

was previously defined by (RUSSAKOVSKY et al., 2015) when they propose to consider a prediction correct, when the target label is within the top 5 class predictions from the Neural Network.

This metric is of high relevance because it has become the benchmark to evaluate models on ImageNet classification tasks. It is the metric used in the works that we use as a comparison to the proposed approaches.

Additionally, this approach can be generalized for a TOP-K accuracy. However, in the scope of the present thesis, we will only use the TOP-5 accuracy as an evaluation metric for the models.

3.2.3 Cumulative FLOPs

FLOPs are an acronym for Floating Point Operations. Floating points have special importance in computation applications since they are the standard way to represent real numbers in modern computers. According to IEE-754 standard (THE INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS, INC, 1985), three elements represent the floating point format.

(1) Sign (Positive/Negative)

(2) Precision (Significant number of real number, mantissa)

(3) Number of digits (Index range)

$$\text{Floating Point Number} = (-1)^{\text{Sign}} \cdot 1.M \cdot 2^{E-(\text{exponent bias})}. \tag{3.2}$$

Being E the binary value of the exponent, S the sign of the number, and M the mantissa, the part of the number after the decimal point.

Deep Neural Networks such as the ones cited in this study contain a high number of parameters in the order of millions. Therefore, a prediction made in one of these networks involves also a large number of floating point operations.

Even though the number of cumulative FLOPs does not necessarily mean a faster and more efficient Neural Network, as shown by (VASU et al., 2023). this number is commonly used as one of the main benchmarks to evaluate the efficiency of a Neural Network in the literature (TAN; LE, 2019). Consequently, throughout this study, it will be also used to compare the efficiency between the studied Networks in the experiments.

3.3 Proposed Approach

This section describes the proposed experiments and their motivation. The results of these experiments are found in chapter 4, and a discussion on the results is in chapter 5.

3.3.1 Motivation

As discussed in chapter 2, there are many works that investigate the role of resolution in the quality of the Convolutional Neural Network output. Even though there are studies showing a direct relation between a higher resolution and a higher accuracy (HAN et al., 2020), there is a lack of investigation in the literature in relation to the proposed research questions.

To investigate the possibility of reducing the resolution of a pre-trained Neural Network while maintaining its predictive ability, we propose a series of experiments, where we apply the resolution reduction on the models, and use the metrics described beforehand to evaluate the models regarding efficiency and accuracy.

3.3.2 Resolution Reduction

Our proposed method consists of copying the weights of a pre-trained Neural Network to a new CNN, with a lower input resolution. It has no additional computational cost, since all the training procedure was already done, which is the step that requires the most processing power.

Despite the simplicity, the described approach has a few limitations. The first is the fact that some architectures are built in a way that it is not possible to reduce the resolution and keep the same structure and weights. For these cases, our method is not possible. Additionally, for this approach to work, the architecture requires a global max pooling layer after the convolutional steps, otherwise, simply transferring the weights to a new Network is also not feasible.

The image sample is also rescaled in order to match the desired resolution, and then the new Neural Network is used to classify the lower-resolution image. It is expected that for low relative resolution differences, there should not be a big discrepancy in the obtained results between the original and the reduced resolution, and that is to be investigated with the experiments.

Figure 6 shows a diagram of the proposed approach for the ImageNet example. The original architecture used for this case uses 224×224 as the input shape, and the image shows an example of reduced resolution 196×196. The weights are copied from the original Network to the new CNN, which is used to predict the class of the re-scaled image.

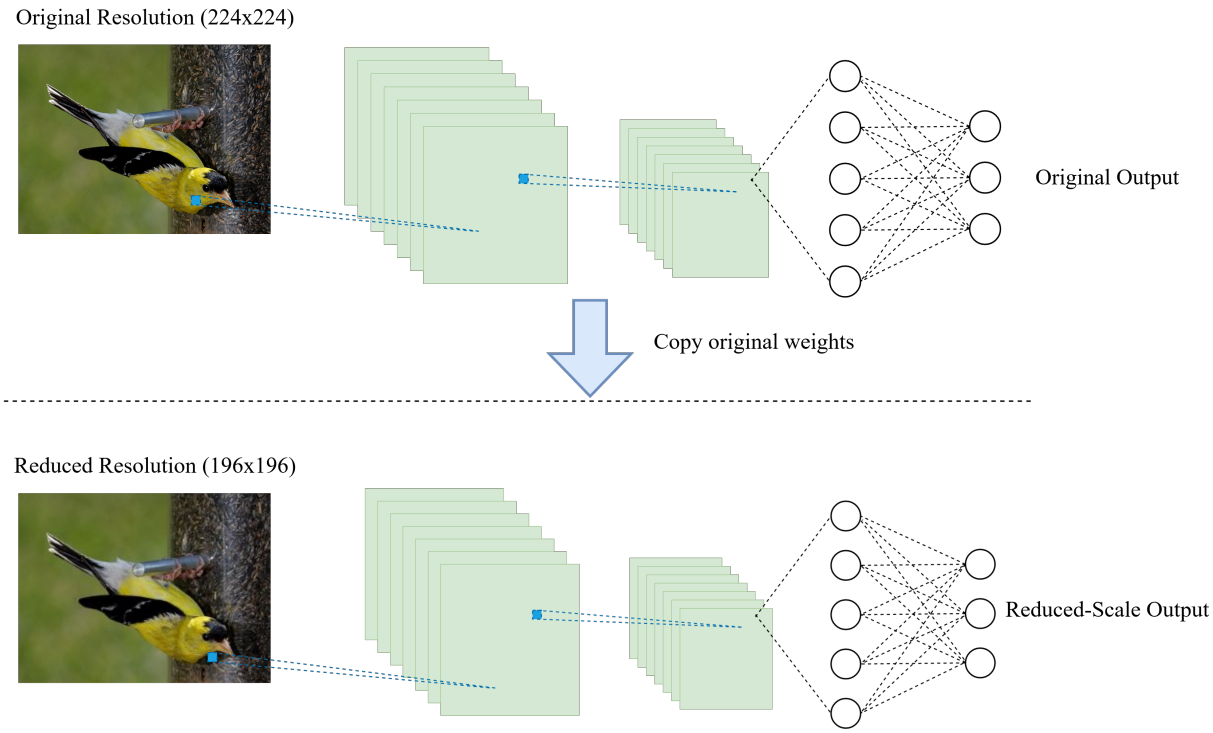

Figure 6 – Reducing resolution method overview

3.3.3 Resolution Selection

Using this proposed method to downscale the Neural Networks resolution, a significant question emerges: Is there a way to define which is the most optimal resolution to be selected for the image to be downscaled? This question is investigated in this thesis based on the following experiment.

3.3.3.1 Random Selection

To investigate this question, we propose a random selection of the resolution, by doing so, there is a high probability of a reduction in the Floating Points Operations numbers, which indicates a more efficient Neural Network model.

The proposed framework consists of randomly selecting a resolution from a set of resolutions for the model to perform the prediction of the image. Figure 7 shows the explanation of this proposed approach for a specific example. In this case, the selector works by choosing randomly from five possible resolutions, including the original one.

Consequently, we hypothesize that the accuracy of the models decreases when the resolution also decreases. If so, the expected accuracy when selecting the resolution randomly is between the original accuracy and the worst accuracy between the selected models.

A similar result is expected for the number of Floating Points Operations since

it decreases when the resolution decreases as well. Consequently, using a simple random resolution selector is expected to reduce the number of FLOPs, at the cost of a quantified reduction on the accuracy.

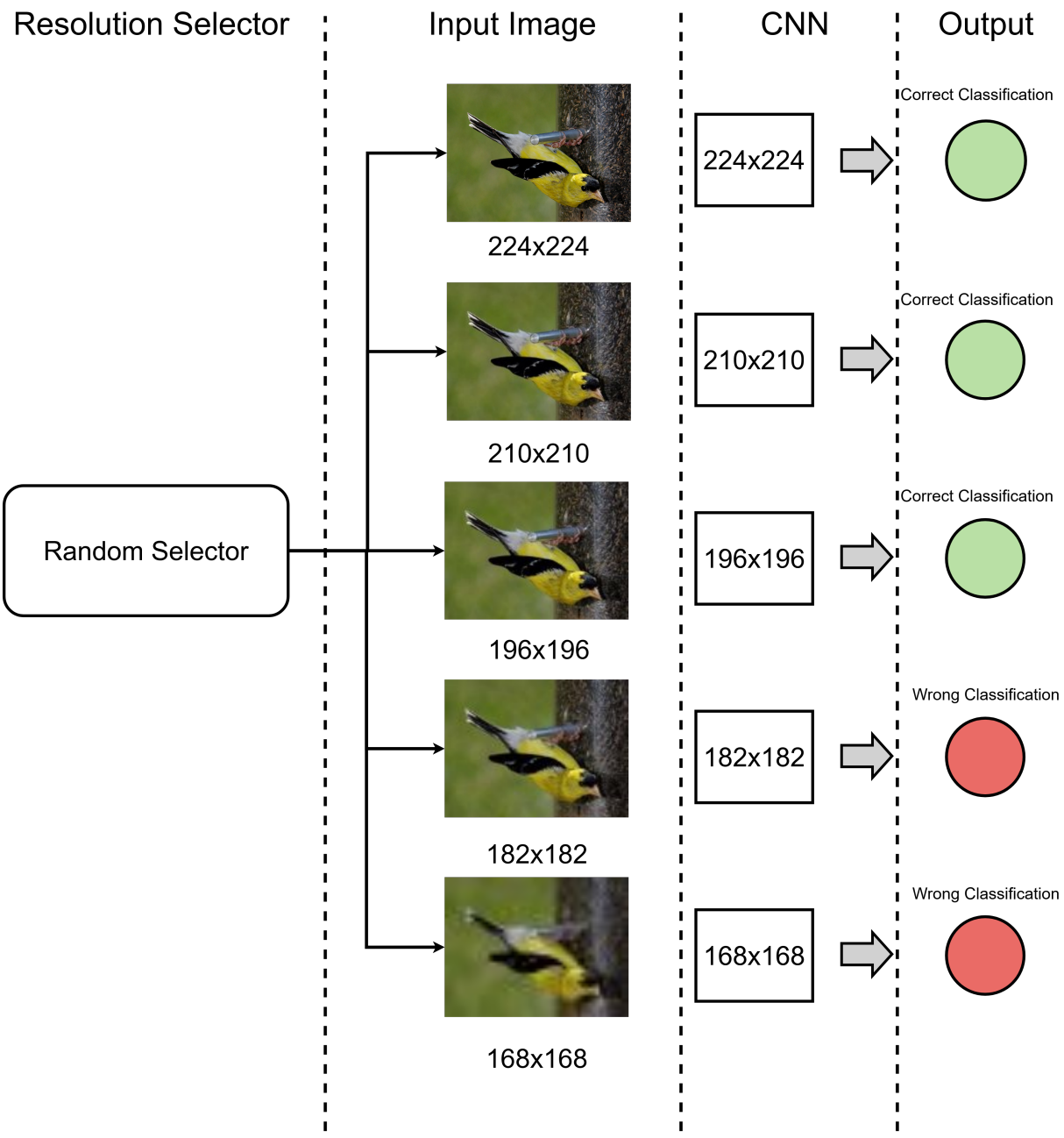

Figure 7 – Reducing

3.4 Model Architectures

To evaluate the proposed approaches, we conducted experiments using different Convolutional Neural Network architectures. The experiments are based on pre-trained models of these different architectures.

More specifically, we use two architectures for the CIFAR-10 dataset, which are the NASNet (ZOPH et al., 2018) and MobileNet V2 (SANDLER et al., 2018). For the ImageNet set, we use three variations of the ResNet architecture (HE et al., 2016). All these architectures are described in more detail throughout this section.

3.4.1 NASNet

The first architecture used in this work is called NASNet (ZOPH et al., 2018), which is an architecture designed with the reinforced learning method called Neural Architecture Search (NAS) (ZOPH; LE, 2017), to optimize model architectures.

The Neural Architecture Search consists of one controller (a RNN) that predicts the architecture A from a search space. The RNN samples child networks that are trained to converge to obtain a certain accuracy on the validation set. Figure 8 shows a diagram of this method

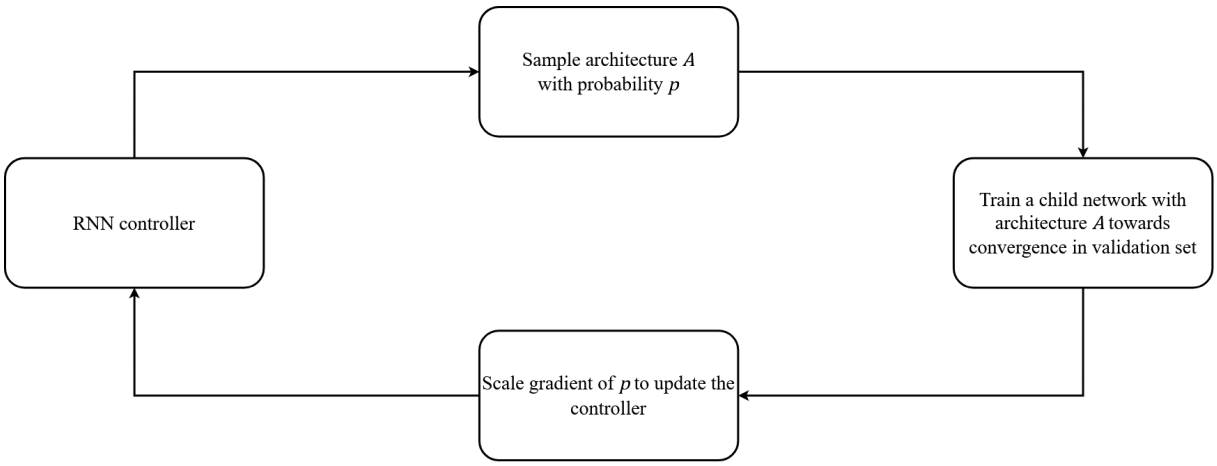

Figure 8 – Neural Architecture Search (NAS) diagram, adapted from (ZOPH et al., 2018)

3.4.2 MobileNet V2

The other architecture we use to perform the CIFAR-10 experiments is the MobileNet V2. The authors of (SANDLER et al., 2018) described an architecture focused on mobile applications based on an inverted residual structure where the shortcut connections are between the thin bottleneck layers.

The design is made based on MobileNetV1 (HOWARD et al., 2017) which uses a technique of splitting the convolutional layer into two separate steps, a depthwise and a pointwise convolution, reducing the computational costs. This split convolutional layer operation is called Dephtwise Separable Convolution.

MobileNetV2 expands this idea, adapting the Depthwise Separable Convolutions into linear bottleneck convolutions, which embed the information contained in a convolu-

tional layer in a lower-dimensional subspace using the ReLU non-linearity. Additionally, residual blocks are added using skip connections.

A more formal description of this approach can be found in (SANDLER et al., 2018), and a diagram with an overview of the MobileNet V1 and V2 separable convolutional blocks is seen in Figure 9.

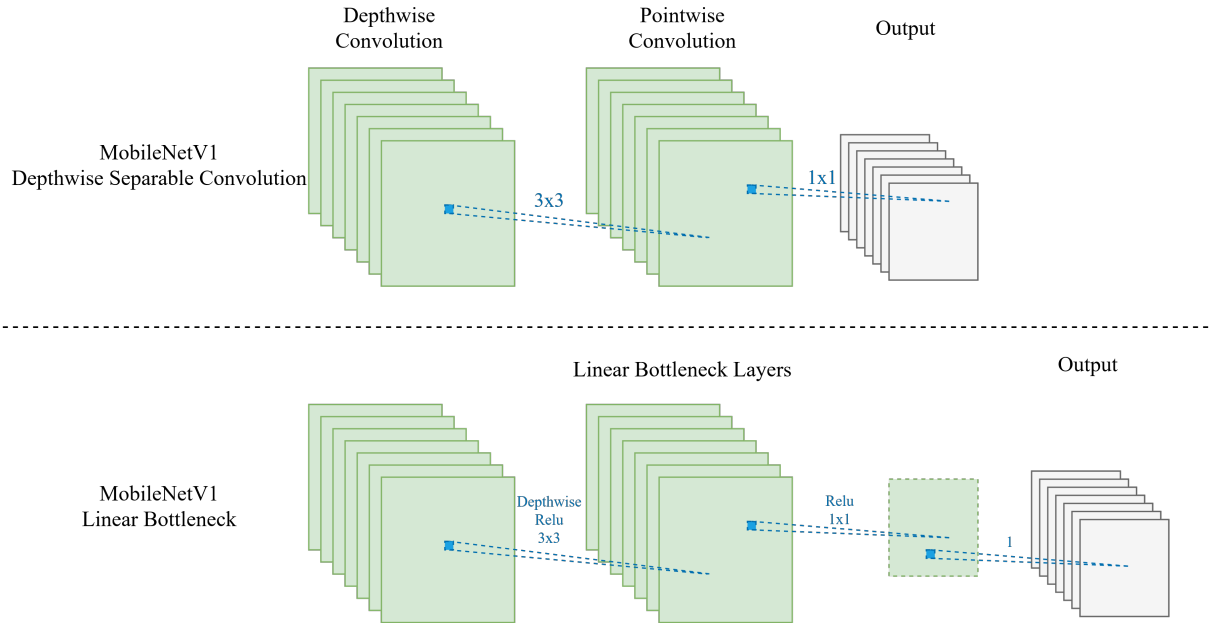

Figure 9 – MobileNet V1 and V2 separable convolution blocks

3.4.3 ResNet

Another architecture we use in this study is the ResNet. The ResNet was first defined in (HE et al., 2016), when, aiming to build deeper Neural Networks, the authors developed a model that restructures the layers as residual functions, to solve the accuracy saturation problem that happened with Deep Networks.

These residual learning functions are mapped using shortcut connections (BISHOP, 1995), which are used to add the output of different layer blocks. Figure 10 shows a diagram of this implementation. This structure is shown to improve the learning for deep Convolutional Networks.

Our work uses three different implementations of ResNet defined in (HE et al., 2016), which are ResNet50, Resnet101, and ResNet152. These are the best-performing architectures on the ImageNet set, and the difference between them is the number of residual blocks within the Network.

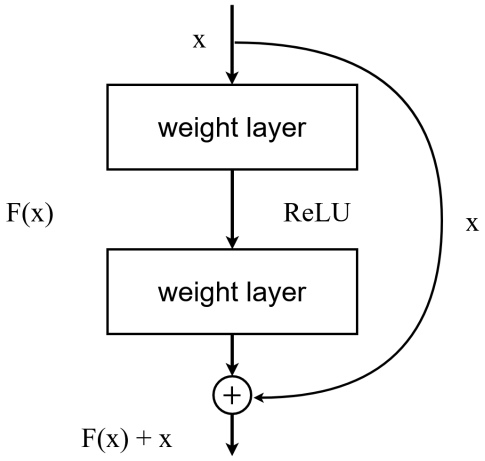

Figure 10 – Residual Learning Block of ResNet. From (HE et al., 2016)

4 Experimental Results

This chapter presents the results achieved during the experiments described in chapter 3. Here, we show only the quantitative results, a discussion on these results is available in chapter 5.

4.1 Resolution Reduction

The first experiments on the CIFAR-10 were made to study the impact of the resolution on the accuracy of low-resolution images. Thereafter, we tested the same reduction of resolution approach on the ImageNet dataset, using ResNet-based architectures to perform the prediction.

4.1.1 CIFAR-10

4.1.1.1 NasNet

Table 1 presents the floating point operations and the percentage point drop in accuracy relative to the original resolution of the CIFAR-10 (32×32) for the NASNet model.

Resolution	**FLOPs Drop (%)**	**Accuracy Drop (%)**
28×28	23.43	1.92
24×24	43.74	4.34
20×20	60.93	13.32
16×16	74.99	29.58
12×12	85.93	52.40

Table 1 – Resolution reduction experiments on the CIFAR-10 using the NASNet architecture

The results of these experiments show that the accuracy quickly degrades for resolutions lower than 20×20, however, especially for the two resolutions closer to the original, there is a significant improvement in the number of operations, and the accuracy is only 4.34% lower than the original one.

4.1.1.2 MobileNet

Furthermore, the experimental results for the CIFAR-10 set using the MobileNet architecture are displayed in Table 2.

Resolution	FLOPs Drop (%)	Accuracy Drop (%)
28×28	7.32	4.50
24×24	31.91	10.95
20×20	37.28	24.67
16×16	74.99	70.29
12×12	78.40	81.83

Table 2 – Resolution reduction experiments on the CIFAR-10 using the MobileNet architecture

Differently from the NASNet results, the results for the MobileNet show an even faster degradation of the model's accuracy. The evaluation on the resolution 28×28 shows an accuracy drop of only 4.34%, however, the number of floating points operations reduction is also low, being only 7%.

For even lower resolutions, the reduction in the operations is more significant, however the number of errors in the. The accuracy drop is 10% higher for the resolution 24×24 and almost 25% for the resolution 20×20, which is significantly worse than NasNet results.

4.1.2 ImageNet

4.1.2.1 ResNet50

Figure 11 displays the Top5 accuracy relative to the input resolution of the Neural Network for the ResNet50. In this image, the original resolution (224×224) is in red. The figure shows only a small degradation of the accuracy for the resolution values between 182 and 210.

Table 3 shows the FLOPs and accuracy drop in percentage in relation to the original resolution of the ImageNet set (224×224) for the ResNet50 model.

Resolution	FLOPs Drop (%)	Accuracy Drop (%)
210×210	3.90	1.22
196×196	14.72	2.90
182×182	29.26	2.70
168×168	38.49	5.13
154×154	49.50	6.57
140×140	57.26	10.72

Table 3 – FLOPs and accuracy reduction by resolution on the ImageNet using the ResNet50 architecture

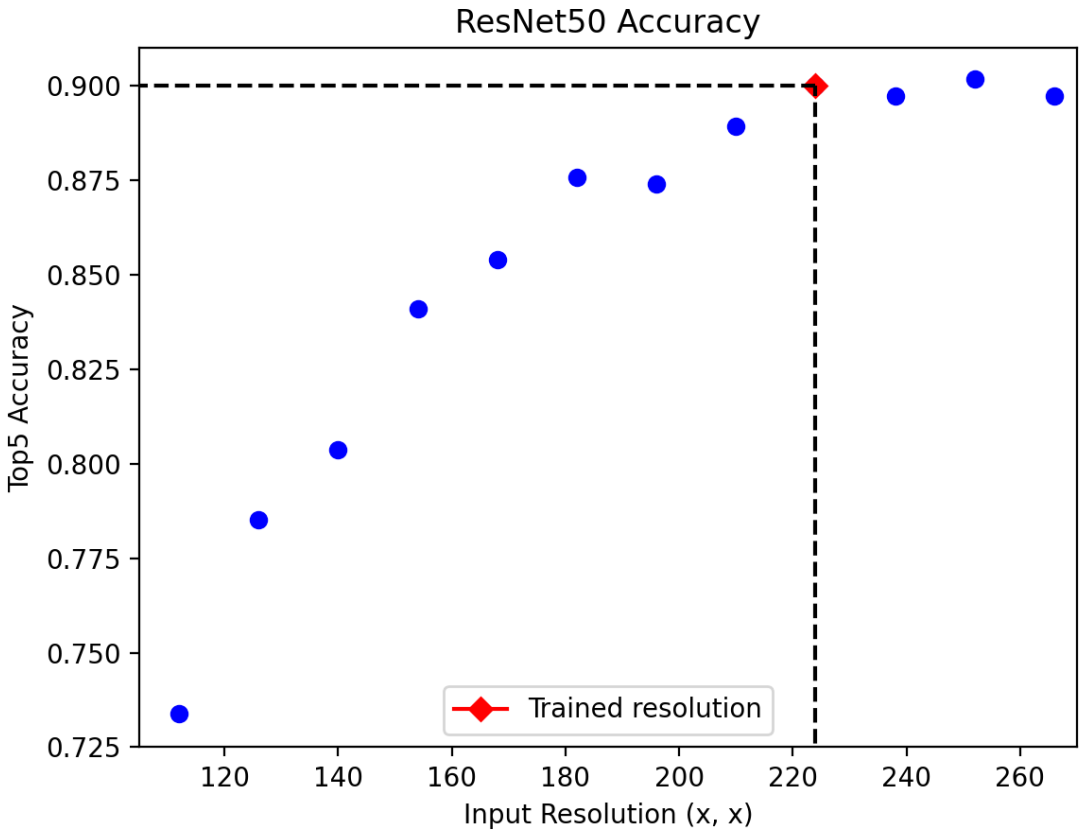

Figure 11 – Accuracy by resolution on the ImageNet using the ResNet50 architecture

4.1.2.2 ResNet101

Similarly to the previous result, Figure 12 shows the TOP5 Accuracy for different input resolution values. The pattern of degradation of the predictive capability is similar to the one seen in ResNet50, where for resolutions closer to the original one, the accuracy drop is low, and for lower reductions in the resolution, the model starts to lose its predictive capabilities.

Another important remark on the figure is the compromise between the resolution and accuracy, which is almost direct for all values below 224×224. However, the model with resolution 182×182 has a higher accuracy than the one with resolution 196×196.

Table 4 shows the reduction in the percentage of the FLOPs and accuracy relative to the original resolution for each evaluated resolution. For the resolution, 168×168, there is a flop reduction of almost 40%, and the accuracy drop is still below 5%.

4.1.2.3 ResNet152

Figure 13 shows the TOP5 accuracy for ResNet152, the last architecture for which the experiments were applied. The figure shows results similar to the previous architectures,

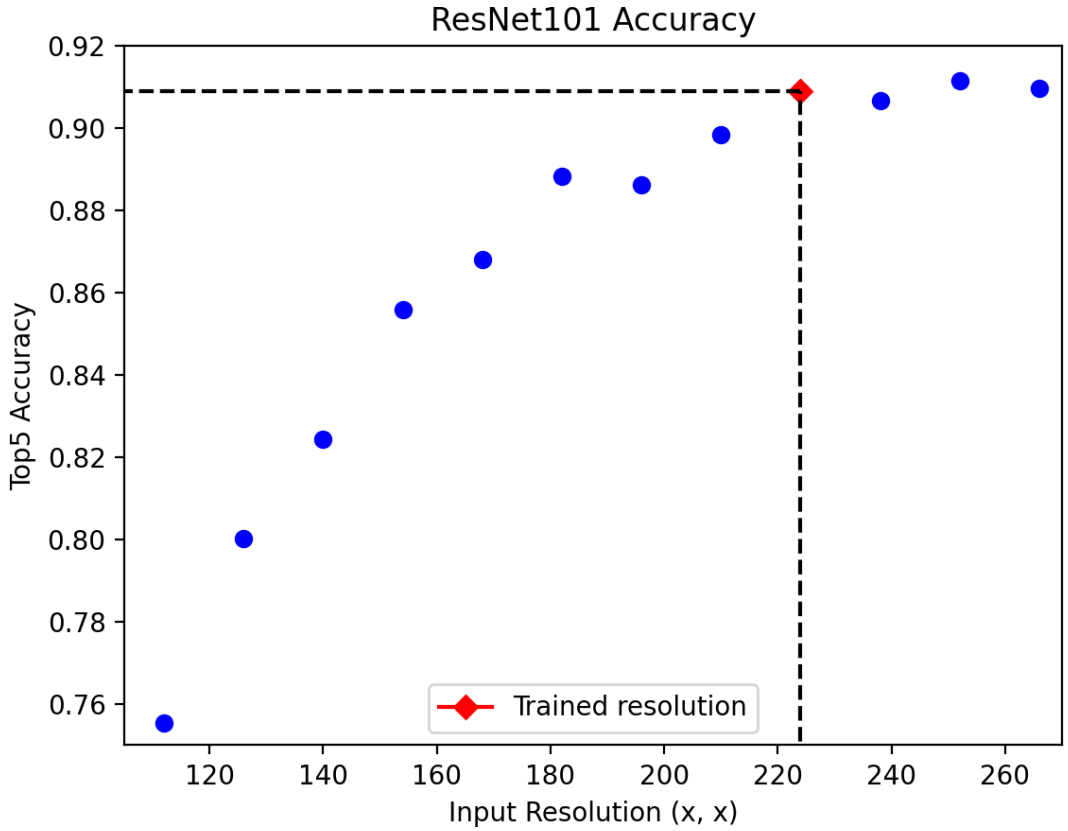

Figure 12 – Accuracy by resolution on ImageNet using the ResNet101 architecture

Resolution	**FLOPs Drop (%)**	**Accuracy Drop (%)**
210×210	1.99	1.16
196×196	14.25	2.50
182×182	27.92	2.29
168×168	38.38	4.50
154×154	49.25	5.86
140×140	57.96	9.31

Table 4 – FLOPs and accuracy reduction by resolution on ImageNet using the ResNet101 architecture

and low reductions for resolution values around the resolution used in training.

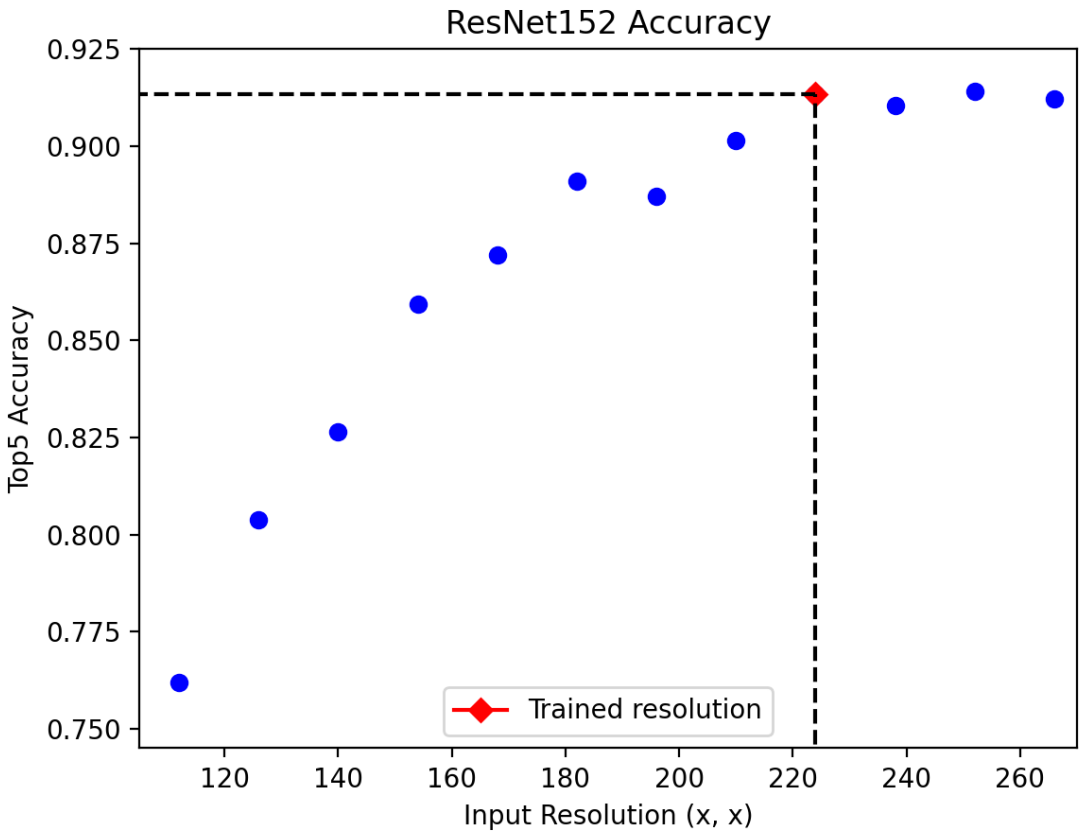

Figure 13 – Accuracy by resolution on the ImageNet using the ResNet152 architecture

The reduction of the FLOPs and accuracy for each resolution is displayed in Table 5, the FLOPs reductions are similar to the ones seen in Table 4, and the accuracy drop is slightly higher for every resolution.

Resolution	**FLOPs Drop (%)**	**Accuracy Drop (%)**
210×210	1.99	1.31
196×196	14.25	2.86
182×182	27.92	2.45
168×168	38.38	4.52
154×154	49.25	5.92
140×140	57.96	9.51

Table 5 – FLOPs and accuracy reduction by resolution on the ImageNet using the ResNet152 architecture

4.2 Random Resolution Selection

Whereas our previous experiments investigate our first research question, the answer to the second research question is examined through the random resolution selection

experiments. The results for these experiments for the three studied architectures can be seen in Table 6. The results presented show the drop in accuracy and FLOPs compared to the model with the original resolution (224 *times*224) for each ResNet architecture.

Architecture	Metric	Average	Standard Deviation	Max	Min
ResNet50	Accuracy Drop (%)	2.42	0.11	2.54	2.24
	FLOPs Drop (%)	17.30	0.04	17.34	17.25
ResNet101	Accuracy Drop (%)	2.09	0.05	2.16	2.02
	FLOPs Drop (%)	16.51	0.03	16.54	16.47
ResNet152	Accuracy Drop (%)	2.04	0.02	2.02	2.06
	FLOPs Drop (%)	16.53	0.04	16.60	16.50

Table 6 – Random resolution selection experiments results for the ResNet50, ResNet101 and ResNet152

The average accuracy reduction for the ResNet50 is 2.42%, which is slightly lower than the reduction obtained using only the resolutions 196×196 and 182×182, and the drop on the FLOPs is on average 17.30 which is higher than the one obtained when using only the resolution 196x196.

For the ResNet101, the average accuracy drop is 2.09, which is again lower than the drop achieved with the resolutions 196×196 and 182×182, and the FLOPs drop is also 16.51%, also higher than the one using only the resolution 196×196.

Finally, the results for the ResNet152 architecture follow the previous ones. The average accuracy drop is 2.04%, a value lower than the drops of the resolutions 196×196 and 182×182 drops. Additionally, the reduction on the floating points operation is on average 16.53%, once more a higher drop than the one of the resolution 196×196, although lower than the FLOPs drop obtained with the resolution 182×182.

4.3 Comparison with State of the Art

To evaluate how the results achieved in this work compare with other FLOPs reduction techniques available in the literature, Table 7 pfdxdngx some literature results for the ResNet50 architecture in contrast with ours.

The table shows different approaches that reduce the FLOPs count with low or no drop in accuracy. The references include models using different techniques, including the pruning of Neural Networks (HU et al., 2016), Dynamic routing using SkipNets (WANG et al., 2018), and model compressing using reinforced learning (HE et al., 2018).

Our results obtained with the ResNet50 architecture are on par with the state-of-the-art results. In particular, for the resolution 182×182 the Floating Points Operations reduction is close to 30%, which is higher than pruning techniques seen in (HU et al., 2016; HE; ZHANG; SUN, 2017) for instance. Additionally, the accuracy drop for this resolution

is only slightly higher than the drop seen in (WANG et al., 2018), with a higher FLOPs reduction.

Resolution	**FLOPs Drop (%)**	**Accuracy Drop (%)**
(HU et al., 2016)	19.69	0.84
(HE; ZHANG; SUN, 2017)	20.00	1.70
(WANG et al., 2018)	20.00	2.00
(HE et al., 2018)	20.00	1.40
(LI et al., 2019)	50.00	0.36
(HE et al., 2019)	53.50	0.55
(GUO; OUYANG; XU, 2020)	55.71	-0.28
(HE et al., 2020)	60.80	0.83
(LUO; WU, 2020)	72.86	1.41
(LIN et al., 2020)	76.03	3.29
Ours (210×210)	3.90	1.22
Ours (196×196)	14.72	2.90
Ours (182×182)	29.26	2.70
Ours (168×168)	38.49	5.13
Ours (154×154)	49.50	6.57
Ours (140×140)	57.26	10.72
Ours (Random Resolution)	17.30	2.42

Table 7 – FLOPs and accuracy reduction by resolution on the ImageNet using the ResNet50 architecture compared to state-of-the-art results

5 Discussion

This chapter discusses in greater depth the results reported in chapter 4. The research questions described in chapter 1 are answered in light of the experiments. Additionally, the contribution of this thesis is understood in comparison to other FLOPs reduction works available in the literature.

5.1 CIFAR-10

As discussed beforehand, the image quality on the CIFAR-10 dataset varies significantly from sample to sample. A high number of objects in the images are occluded or unclear, therefore, even humans make mistakes when identifying them, and this fact together with the low resolution of the image makes the challenge of classification a difficult task by itself. Thus, it is expected that reducing the resolution even more should have a high impact on the accuracy.

Consequently, the results seen in subsection 4.1.1 do not come as a surprise, reducing the resolution from 32×32 to 28×28 reduces the accuracy of the model already by 4.34%. Furthermore, an even bigger reduction of the resolution degenerates the Neural Network prediction capability completely. Resolutions below 20×20 show a drop in accuracy of 29.58%.

Given the discussed limitations of the dataset and the low resolutions of the available images, it is possible to infer that our proposed solution does not work for low-resolution networks. The initially proposed research question is therefore answered this way: It is not possible to keep the predictive capability of pre-trained CNNs when reducing its resolution when the resolution of training is already low.

Because of a negative answer to the first question, the second research question could not be investigated for the CIFAR-10 dataset. Therefore, we leave this investigation for future research.

5.2 ImageNet

5.2.1 Resolution Reduction

The experiments with resolution reduction achieved interesting results. For all three studied architectures, the reduction in accuracy was 5.13% for the resolution 168×168. Also, we achieved a reduction in the number of floating point operations as high as 29.26%

for accuracy drops lower than 2.70%.

When comparing these results to the literature in Table 7, we confirm that our solution brings a FLOPs reduction comparable to the other compressing techniques available, at an expense of a drop in accuracy that is not much higher than the ones seen in other results.

Even though not optimal, our solution has a big advantage: it does not involve any additional cost on top of the model training itself. The approach consists of simply copying the weights from the original resolution to a lower, re-scaled resolution. This means that it can be easily reproduced and does not require additional training steps.

These experiments, consequently, answer affirmatively to the main research question proposed beforehand. It is indeed possible to reduce the resolution of a pre-trained Neural Network and keep its predictive capability. The accuracy of the models reduces with the resolution reduction, but for varied resolution values, the drop in accuracy is not high.

5.2.2 Random Resolution

The accuracy drop obtained when choosing the resolution randomly is, on average, not much better than other reduction methods. However, the average reduction in FLOPs is lower than all the other studies used for comparison.

On the one hand, the advantage is similar to the one discussed in the previous section: the proposed approach does not have any additional computational cost for a pre-trained Neural Network. Using this random resolution selector, we only copy the weights from the original Neural Network to a randomly chosen resolution, rescale the input image to the same resolution, and then evaluate the data. Even this simple method is shown to achieve positive performance. On the other hand, the FLOPs reduction when using a random resolution is low compared to other methods. In fact, for the ResNet50 architecture, we achieve a better reduction in the number of floating points operations and for only a slightly higher drop in accuracy using a fixed 168×168 resolution.

This result by itself indicates that developing a method that can systematically define which is the better resolution can potentially bring better outcomes regarding the FLOPs and accuracy tradeoff than just selecting the resolution randomly.

Therefore, the answer to the research question can be answered positively, and a random selection of the resolution of the CNN is promising. Choosing the resolution randomly leads to a reduction in the floating points operations, with a quantified loss in accuracy. However, the results achieved for these experiments indicate that there is space for a more systematic method to select the resolution which could yield better results.

5.3 Remarks

One important remark on the proposed approach is that for it to be successfully applied, the Convolutional Neural Network needs to fulfill a small set of requirements.

The first one is that the architecture has to support the scale reduction, for it to be implemented. It means that if the model's architecture is designed in order to support only specific resolutions, or has a minimum resolution equal to the training resolution, it is not possible to apply the techniques described in this thesis.

Additionally, the other requirement is that the architecture also needs a global max pooling layer (SUDHOLT; FINK, 2016; OQUAB et al., 2015). Gladly, modern Convolutional Networks architectures often have a global max pooling layer before the fully connected classification layers. This is the case, especially for all the architectures studied in this thesis (ZOPH; LE, 2017; SANDLER et al., 2018; HE et al., 2016).

6 Conclusion

This concluding chapter serves as the end of the thesis, offering a general perspective on the accomplishments and contributions of the work. We also discuss possibilities for future research on the topic.

6.1 Thesis Summary and Conclusion

This thesis investigated the research questions proposed in the chapter 1. In particular, the first question inquires about the possibility of reducing the resolution of a pre-trained Neural Network while keeping its predictive ability. Whereas the second question explores the possibility of systematically determining a resolution for which the model keeps its predictive capacity.

The first step was to investigate other relevant works in the literature that study the role of the resolution on Neural Networks. This step provided an overview of the impact of changing the resolution of the models and an insight that the answer to the first question could potentially be positive.

To investigate the possibility of reducing the resolution of pre-trained Convolutional Neural Networks while maintaining their predictive abilities, we conducted experiments with different architectures for two different datasets: CIFAR-10 trained with a low resolution (32×32), and the ImageNet, trained with a 224×224 resolution.

The experiments were made by transferring the weights from the original model to models with different resolutions and evaluating the accuracy and FLOPs count of these new models for re-scaled images. The next step was to investigate the question about resolution selection. The goal is to understand the possibility of a method that selects the resolution to be used by the Convolutional Network to perform the predictions.

Consequently, experiments were set up to evaluate the results when selecting the model's architecture randomly. Again, the accuracy and FLOPs reduction compared to the original resolution are evaluated. We discussed the acquired results in light of the original proposed questions. The first conclusion is that, for Networks trained in higher resolutions, it is indeed possible to reduce the resolution of the network and reduce considerably the floating points operations number, at the cost of only a small drop in accuracy.

Lastly, we concluded that it is possible to define a systematic method to select the resolution. Our experiments selecting the resolutions randomly show a FLOPs reduction comparable to other approaches found in literature, with a low reduction in accuracy. The work also indicates the possibility of defining a more refined method to select the Neural

Network scale.

6.2 Future Work

As a future work, we propose more experiments to continue investigating the research questions, and their applicability in other environments. The proposed methods can potentially be diversified for more datasets and adapted for different architectures. Thus, we propose to investigate the impact of resolution on different architectures, including more modern networks such as attention-based-classifiers (DOSOVITSKIY et al., 2020; DAI et al., 2021). Additionally, we propose to study the impact of the resolution on segmentation tasks. In particular, applying the resolution reduction technique proposed here on U-Net-based architectures for image segmentation (RONNEBERGER; FISCHER; BROX, 2015).

As for the second research question, the present work raised the possibility of a more optimal method to select the resolution for the Convolutional Network. We propose, therefore, in future work, to study and evaluate different systematic resolution selectors, and compare them to the random selector proposed here.

Bibliography

ABIYEV, R. H.; ARSLAN, M.; IDOKO, J. B. Sign language translation using deep convolutional neural networks. *KSII Transactions on Internet & Information Systems*, v. 14, n. 2, 2020. Cited in page 11.

AHMED, W. S.; KARIM, A. a. A. The impact of filter size and number of filters on classification accuracy in cnn. In: *2020 International Conference on Computer Science and Software Engineering (CSASE)*. [S.l.: s.n.], 2020. p. 88–93. Cited in page 11.

BISHOP, C. M. *Neural networks for pattern recognition*. [S.l.]: Oxford university press, 1995. Cited in page 30.

DAI, Z. et al. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, v. 34, p. 3965–3977, 2021. Cited in page 46.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255. Cited in page 23.

DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations*. [S.l.: s.n.], 2020. Cited in page 46.

FENG, X. et al. Computer vision algorithms and hardware implementations: A survey. *Integration*, Elsevier, v. 69, p. 309–320, 2019. Cited in page 19.

GALVEZ, R. L. et al. Object detection using convolutional neural networks. In: IEEE. *TENCON 2018-2018 IEEE Region 10 Conference*. [S.l.], 2018. p. 2023–2027. Cited in page 11.

GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. <http://www.deeplearningbook.org>. Cited 2 times in pages 19 and 20.

GUO, J.; OUYANG, W.; XU, D. Multi-dimensional pruning: A unified framework for model compression. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 1508–1517. Cited in page 39.

HAN, K. et al. Model rubik's cube: Twisting resolution, depth and width for tinynets. *Advances in Neural Information Processing Systems*, v. 33, p. 19353–19364, 2020. Cited 4 times in pages 11, 15, 16, and 26.

HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 770–778. Cited 5 times in pages 11, 29, 30, 31, and 43.

HE, Y. et al. Learning filter pruning criteria for deep convolutional neural networks acceleration. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2020. p. 2009–2018. Cited in page 39.

HE, Y. et al. Amc: Automl for model compression and acceleration on mobile devices. In: *Proceedings of the European conference on computer vision (ECCV)*. [S.l.: s.n.], 2018. p. 784–800. Cited 2 times in pages 38 and 39.

HE, Y. et al. Filter pruning via geometric median for deep convolutional neural networks acceleration. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 4340–4349. Cited in page 39.

HE, Y.; ZHANG, X.; SUN, J. Channel pruning for accelerating very deep neural networks. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2017. p. 1389–1397. Cited 2 times in pages 38 and 39.

HO-PHUOC, T. Cifar10 to compare visual recognition performance between deep neural networks and humans. *arXiv preprint arXiv:1811.07270*, 2018. Cited in page 23.

HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. Cited 3 times in pages 11, 15, and 29.

HU, H. et al. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016. Cited 2 times in pages 38 and 39.

KRIZHEVSKY, A.; HINTON, G. et al. Learning multiple layers of features from tiny images. Toronto, ON, Canada, 2009. Cited in page 23.

LI, J. et al. Oicsr: Out-in-channel sparsity regularization for compact deep neural networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 7046–7055. Cited in page 39.

LI, Q. et al. Medical image classification with convolutional neural network. In: IEEE. *2014 13th international conference on control automation robotics & vision (ICARCV)*. [S.l.], 2014. p. 844–848. Cited in page 11.

LIN, M. et al. Hrank: Filter pruning using high-rank feature map. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2020. p. 1529–1538. Cited in page 39.

LUNDERVOLD, A. S.; LUNDERVOLD, A. An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*, Elsevier BV, v. 29, n. 2, p. 102–127, maio 2019. Disponível em: <https://doi.org/10.1016/j.zemedi.2018.11.002>. Cited in page 17.

LUO, J.-H.; WU, J. Neural network pruning with residual-connections and limited-data. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 1458–1467. Cited in page 39.

OQUAB, M. et al. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 685–694. Cited in page 43.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. [S.l.], 2015. p. 234–241. Cited in page 46.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, Springer Science and Business Media LLC, v. 323, n. 6088, p. 533–536, out. 1986. Disponível em: <https://doi.org/10.1038/323533a0>. Cited in page 19.

RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, Springer, v. 115, p. 211–252, 2015. Cited in page 25.

SANDLER, M. et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 4510–4520. Cited 3 times in pages 29, 30, and 43.

SCHERER, D.; MüLLER, A.; BEHNKE, S. Evaluation of pooling operations in convolutional architectures for object recognition. In: *Artificial Neural Networks – ICANN 2010*. Springer Berlin Heidelberg, 2010. p. 92–101. Disponível em: <https://doi.org/10.1007/978-3-642-15825-4_10>. Cited in page 22.

SUDHOLT, S.; FINK, G. A. Phocnet: A deep convolutional neural network for word spotting in handwritten documents. In: IEEE. *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. [S.l.], 2016. p. 277–282. Cited in page 43.

TAN, M.; LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR. *International conference on machine learning*. [S.l.], 2019. p. 6105–6114. Cited 3 times in pages 11, 15, and 25.

THAMBAWITA, V. et al. Impact of image resolution on deep learning performance in endoscopy image classification: an experimental study using a large dataset of endoscopic images. *Diagnostics*, MDPI, v. 11, n. 12, p. 2183, 2021. Cited in page 16.

THE INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS, INC. *IEEE Standard for Binary Floating-Point Arithmetic*. [S.l.], 1985. Cited in page 25.

TOUVRON, H. et al. Fixing the train-test resolution discrepancy. *Advances in neural information processing systems*, v. 32, 2019. Cited in page 16.

VASU, P. K. A. et al. Mobileone: An improved one millisecond mobile backbone. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2023. p. 7907–7917. Cited 2 times in pages 15 and 25.

WANG, X. et al. Skipnet: Learning dynamic routing in convolutional networks. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 409–424. Cited 2 times in pages 38 and 39.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.], 2014. p. 818–833. Cited in page 21.

ZHANG, X. et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 6848–6856. Cited in page 15.

ZOPH, B.; LE, Q. Neural architecture search with reinforcement learning. In: *International Conference on Learning Representations*. [s.n.], 2017. Disponível em: <https://openreview.net/forum?id=r1Ue8Hcxg>. Cited 2 times in pages 29 and 43.

ZOPH, B. et al. Learning transferable architectures for scalable image recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 8697–8710. Cited in page 29.