

Noora Sassali

DEEP LEARNING WITH FOURIER TRANSFORMED IMAGES

Faculty of Information Technology and Communication Sciences

Bachelor's thesis

May 2023

ABSTRACT

Noora Sassali: Deep learning with Fourier transformed images
Bachelor's thesis
Tampere University
Bachelor's Programme in Computing and Electrical Engineering
May 2023

The research and applications based on deep learning have increased rapidly over the past years. Image classification is one of the main applicational scopes of deep learning. Many existing image classifiers are based on Convolutional Neural Network (CNN) architectures.

Discrete Fourier Transformation (DFT) is a powerful tool in image processing and can be used to analyze 2D data and perform filtering in the frequency domain. DFT has a few applications in deep learning, such as a spectral pooling layer. Fourier transformation is a frequently used pre-processing method in audio deep learning, as the learning benefits from the signal representation in the frequency domain.

This thesis evaluates how a CNN backbone learns an image classification task if training is performed with Fourier-transformed images instead of spatial RGB images. Two experimental models based on EfficientNetV2-S architecture were created for DFT pre-processed and spatial RGB images. A pre-trained version of the backbone was used as the baseline model. ImageNet 2012 dataset reduced to 64x64 resolution was used as the training and testing data. DFT and RGB models were trained from scratch and evaluated by comparing model accuracies (top-1, -2, -5).

The results suggest that CNN can learn features from both RGB and Fourier-transformed images. However, the spatial image network reaches higher accuracy over the trained epochs. Model accuracies are significantly smaller when compared to the baseline but could be improved by fine-tuning the training process further and using a larger version of the ImageNet dataset. Further studies are suggested to determine what inputs benefit DFT pre-processing the most. Research could be extended to exploit Fourier Transformation properties in CNN architectures further.

Keywords: deep learning, convolutional neural networks, image classification, discrete fourier transformation, frequency domain

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

TIIVISTELMÄ

Noora Sassali: Syväoppiminen Fourier-muunnetuilla kuvilla
Kandidaatintyö
Tampereen yliopisto
Tieto- ja sähkötekniikan kandidaattiohjelma
Toukokuu 2023

Neuroverkkoihin perustuvan syväoppimisen (eng. deep learning) tutkimus ja sovellukset ovat kasvattaneet suosiotaan viime vuosien aikana. Kuvien luokitus on yksi syväoppimisen pääkäyttöaloista ja monet olemassa olevista luokittelijoista perustuvat konvoluutioneuroverkkorakenteisiin.

Diskreetti Fourier-muunnos (DFT) on tehokas työkalu kuvankäsittelyssä. Sitä voidaan käyttää kaksiulotteisen datan analysointiin ja erilaisten taajuustasossa tapahtuvien suodatusten toteuttamiseen. Menetelmällä on muutamia sovelluksia kuvien luokitukseen keskittyvässä syväoppimisessä. Sitä käytetään myös syväoppimisen audiosovelluksissa datan esikäsittelymenetelmänä, sillä oppiminen hyötyy äänisignaalien taajuustason kuvauksesta.

Tämän kandidaatintyön tarkoituksena oli arvioida, miten konvoluutioverkkoihin perustuva neuroverkkoarkkitehtuuri oppii kuvien luokitusta, jos kuvasyötteet on muunnettu Fourier-muunnoksen avulla tilatasosta taajuustasoon. Kaksi kokeellista EfficientNetV2-S arkkitehtuuriin perustuvaa verkkoa luotiin sekä DFT-esikäsittelylle että RGB-kuville. Vertailukohtana hyödynnettiin vastaavan verkon esiopetettua versiota. Opetusdatana käytettiin ImageNet 2012 kuva-aineiston 64x64 resoluutioista versiota, joka sisältää 1,2 miljoonaa kuvaa ja 1000 luokkaa. DFT ja RGB-mallit opetettiin alustamattomista verkoista ja niiden suoriutumista arvioitiin mallien tarkkuuksien (top-1,-2 ja -5) avulla.

Työn tulokset osoittavat, että konvoluutioverkko pystyy oppimaan piirteitä sekä RGB- että Fourier-muunnetuista kuvista. Tilatason kuvista oppiva malli saavuttaa kuitenkin DFT-mallia korkeamman tarkkuuden opetusiteraatioiden tuloksena. Mallien saavuttamat tarkkuudet jäivät huomattavasti pienemmäksi esiopetettuun verkkoon verrattuna, mutta niitä voitaisiin parantaa hienosäätämällä oppimisprosessia pidemmälle ja käyttämällä ImageNet kuva-aineiston suurempaa versiota. Tulevaisuudessa voitaisiin tutkia millaiset kuvat hyötyvät Fourier-muunnoksesta eniten ja voidaanko Fourier-muunnoksen erityisominaisuuksia hyödyntää osana konvoluutioverkkoarkkitehtureja.

Avainsanat: syväoppiminen, konvoluutioverkot, kuvien luokitus, diskreetti Fourier-muunnos, taajuustaso

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

CONTENTS

1. Introduction	1
2. Related works	3
3. Methodology	5
3.1 Convolutional Neural Networks	5
3.2 EfficientNetV2	7
3.2.1 Mobile inverted Bottleneck operators	7
3.2.2 Squeeze-and-Excitation blocks	9
3.3 Discrete Fourier Transformation	10
4. Experimental setup	13
4.1 Training setup	14
4.1.1 Optimizer: Stochastic Gradient Descent	14
4.1.2 Loss criterion: Cross Entropy	15
4.2 Evaluation Metrics	16
5. Results and Analysis	17
5.1 Accuracy and loss development	17
5.2 Class-wise accuracy	18
6. Conclusion	21
References	23

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	Artificial Neural Network
CE	Cross Entropy
CNN	Convolutional neural network
DFT	Discrete Fourier Transformation
DNN	Deep Neural Network
FC	Fully connected layer
FFT	Fast Fourier Transformation
FLOPs	Floating point operations; how many operations are required to run a single instance of a given model
GD	Gradient Descent
GPU	Graphical Processing Unit
IDFT	Inverse Discrete Fourier Transformation
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
ReLU	Rectified Linear Unit, activation function
RGB	The color model to represent colors on a display screen using red, green and blue light
SE	Squeeze-and-Excitation block
SGD	Stochastic Gradient Descent
SOTA	state-of-the-art
STFT	Short-Time Fourier Transformation

1. INTRODUCTION

Recent advances in machine learning have drawn media attention to applications based on deep learning. An application of conversational artificial intelligence, ChatGPT, revolutionized the way information can be searched and is likely to affect the field of scientific research [1]. Deep-learning-powered object detection and recognition play a vital role in the rapid progress of the self-driving vehicle industry [2]. Hyper-realistic deep fake videos raise ethical concerns in the era of disinformation [3].

Deep learning is a field of machine learning based on artificial neural networks (ANN), in which multiple layers of processing are used to extract progressively higher-level features from data [4]. Neural networks can produce results that only a few years ago were unattainable, and the given examples are merely the tip of the ice peak. Published documents associated with deep learning have increased each year steadily [5] after the re-discovery of Convolutional Neural Networks (CNN) in 2012. The field has extended over many industries and keeps evolving rapidly.

Image classification is one of the main applicational scopes of deep learning. Training a neural network for a classification task is commonly implemented as supervised learning, using a neural network architecture and large image datasets. The training process includes steps for pre-processing the images before feeding them into the network. Research has shown that data augmentation can have a crucial effect on learning [6].

Fourier theorem is used in various fields such as communication, signal, and image processing. In image processing, Discrete Fourier Transformation (DFT) is a powerful tool to analyze 2D data and perform filtering in the frequency domain. DFT and its inverse can create efficient pooling layers in CNN architectures [7]. On the other hand, Short-time Fourier Transformation is a typical pre-processing method for DNNs involving audio data [8]. Can image-classifying neural network benefit from the frequency representation?

This thesis evaluates how well a CNN backbone learns image classification with DFT pre-processed images compared to spatial RGB images. Two separate backbone networks are trained from scratch, the first using spatial RGB images and the second using DFT pre-processed images. Model accuracies are calculated, and the results are compared to the baseline network, the backbone network with pre-trained weights. The experiment process is illustrated in Figure 1.1.

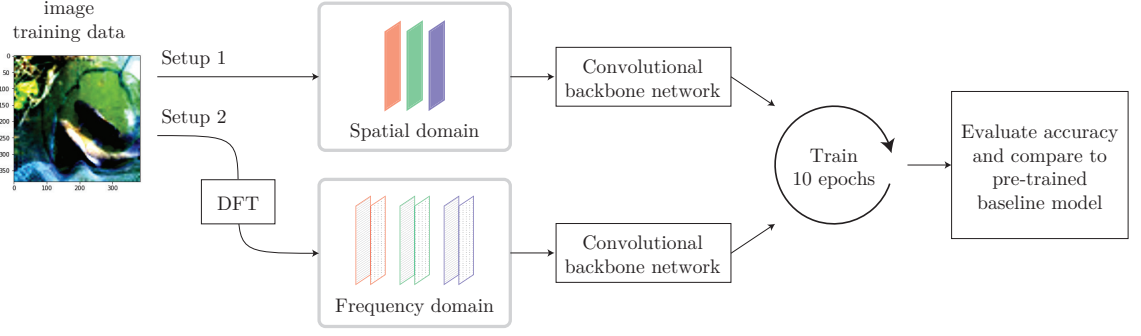


Figure 1.1. The experiments train two Convolutional Neural Network (CNN) based models. The first network is trained with spatial RGB data, and the second with Discrete Fourier Transformed (DFT) data in the frequency domain. Sample image from ImageNet [9].

The remainder of the paper is organized as follows. Section 2 gives an overview of related works. Section 3. presents the used backbone neural network and how discrete Fourier transformation is implemented for images. Section 4 introduces the experiment setups and evaluation metrics in detail. The results and analysis are delivered in Section 5. Finally, Section 6 concludes the work by discussing the experiments and ideas for future work.

2. RELATED WORKS

Neural networks have significantly evolved since the earliest McCulloch-Pitts neuron [10] in 1943 and the Perceptron model of Rosenblatt [11][12] in the 1950s. Within the scope of image classification tasks, a breakthrough in the field happened in 2012, when Krizhevsky et al. [13] revived the concept of convolutional neural networks (CNN) and established record-breaking results in ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The research on deep learning and CNNs increased rapidly [5], and new CNN-based architectures and techniques have emerged ever since.

Image classifying state-of-the-art (SOTA) architectures are often evaluated with the benchmarking ImageNet Challenge. In 2015, Simonyan and Zisserman [14] presented the VGG-16 network family, which improved AlexNet results significantly by increasing the depth of the network, using smaller 3x3 convolutional filters, and applying other improvements. The residual learning framework by He et al. [15] extended the depth further, solving the degradation problem of deeper models with residual mapping. The ResNet family was the first to include shortcut connections to perform identity mapping.

In 2018, NASNets by Zoph et al. [16] presented an improved way to design CNN architectures. Inspired by NAS framework [17], NASNets used reinforcement learning successfully to optimize architecture configurations, surpassing earlier SOTA performances. By contrast, Touvron et al. [6] studied different data augmentation methods, fine-tuning model training and reaching better performance. The latest SOTA architectures, such as ViT-G/14 [18] and BASIC-L [19], are based on Vision Transformers instead of CNNs. Within the scope of this thesis, the most relevant network family is CNN-based EfficientNetV2 [20].

Discrete Fourier Transformation has few applications in image-classifying neural networks besides its wide application in image processing. The work of Ye and Chen [21] presents a Fourier convolutional neural network (F-ConvNet) for human detection and activity classification tasks. The network takes raw frames of radar data as input, which is enframed in the shape of a 2D array. The Fourier layer contains two branches for real and imaginary parts and is trained using Cross Entropy loss and Fourier kernel initializations. [21]

The work of Rippel et al. [7] proposes a spectral pooling layer and parametrization for CNN architectures based on DFT. Spectral pooling enabled pooling to desired output di-

dimensionality while retaining more information than other pooling approaches. The dimensionality reduction is performed by truncating the representation in the frequency domain. Fourier functions were also shown to provide the basis for filter parametrization in CNNs. [7]

Fourier transformation is often used in deep learning applications of audio signal processing. Short-time-Fourier transformation (STFT) can be used as a pre-processing method to create a spectral representation of audio data. Deep learning models can, e.g., model the spectral structure of signal sources by predicting a separation mask based on the mixture input in STFT form. STFT can be efficiently implemented and easily inverted. [8]

3. METHODOLOGY

After the renaissance of CNNs in 2012, research on deep learning and CNNs increased rapidly [5]. The research focus shifted from layer-level details to developing new architectures while harnessing CNN's ability to extract features. Many proposed architectures became well-known models addressing different problems, such as computer vision tasks. Backbone networks are pre-trained versions of these recognized architectures.

Backbones can be a powerful tool to apply deep learning further and design new architectures. Choosing a suitable backbone is essential when preparing a network for a specific task [22, p. 2]. Pre-training itself offers a significant advantage over neural networks trained from scratch. The network weights can be fine-tuned by selecting only a few layers for further training, such as the last full-connected classifying layers. When only a portion of the original parameters is re-initialized and trained, the training is faster and requires less computational power.

This thesis uses EfficientNetV2-S [20] backbone to compare the learning process of two CNNs in an image classification task. The first setup takes spatial RGB images as an input and the second DFT pre-processed images. The pre-trained backbone was used as a baseline model for the experiments. Section 3.1 gives a brief overview of a typical CNN structure, while Section 3.2 describes EfficientNetV2-S architecture and its relevant building blocks in detail. Section 3.3 presents the theory behind 2D Fourier transformation.

3.1 Convolutional Neural Networks

Figure 3.1 illustrates a typical CNN structure with three stages. In the first stage, convolution kernels are slid over the 2D image to compute the convolution or cross-correlation over the spatial neighborhood. An example of convolution as matrix multiplication is shown in Figure 3.2. The process typically takes full advantage of GPU and is implemented in parallel. Each kernel produces a set of linear activations run through a non-linear activation function in the second, *detector*, stage. Lastly, the created feature maps are pooled to modify the output function.[23, p. 335]

In comparison to earlier fully connected Multilayer Perceptron (MLP) networks, CNN is

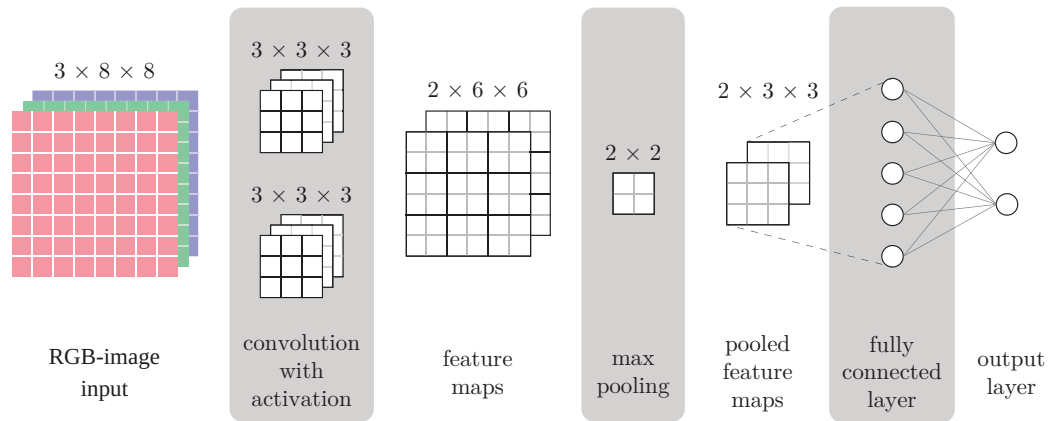


Figure 3.1. An example of simple Convolutional neural network (CNN) setup. Convolving the original RGB input with two kernels creates two feature maps. The maps are pooled by taking the maximum value of each 2x2 area, and the resulting maps are fully connected to the final classifying layers.

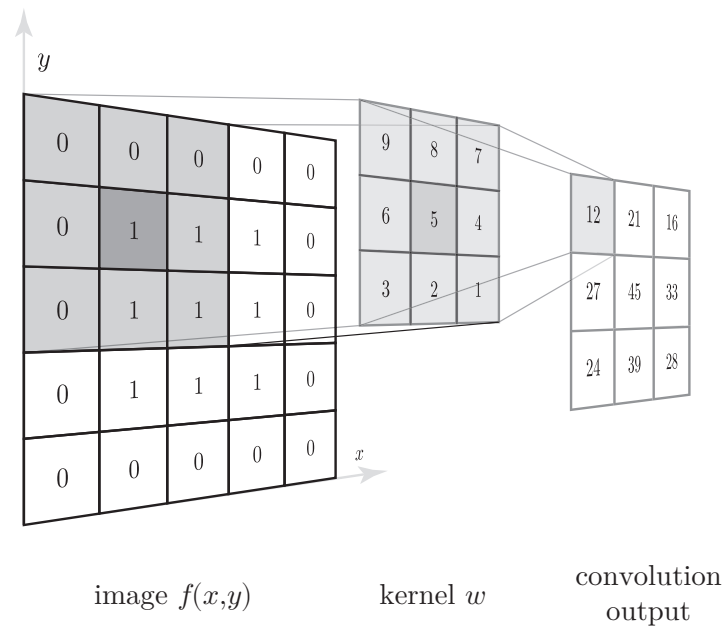


Figure 3.2. Convolution over an image, with a kernel size 3x3. In mathematical terms, the operation used in CNN layers often equals cross-correlation. The parameters of the convolutional layer consist of kernel weights and an additional bias.

capable of learning 2D features directly from raw image data [24, p.964–965]. The convolutional layers have shared parameters consisting of kernel weights and bias. The amount of needed parameters is less compared to fully connected layers, which improved the computational efficiency significantly. The shared parameters give the layer a property called equivariance to translation, while the pooling function helps to make the representation invariant to small translations of the input [23]. These attributes made CNNs

efficient feature extractors still used in state-of-the-art neural networks in image classification.

3.2 EfficientNetV2

EfficientNets is a family of convolutional networks first presented by Tan and Le in 2020. EfficientNets improved the accuracy and efficiency of the previous convolutional networks through a compound scaling method, answering the need for efficiency in previous over-parameterized CNNs. The method used fixed scaling coefficients to uniformly scale the network width, depth, and resolution. The created models were optimized for Floating Point Operations Per Second (FLOPS) and parameter efficiency. [25]

The first EfficientNet architecture, EfficientNet-B0, closely follows the architecture of MnasNET [26]. The architecture was successfully scaled up with a compound scaling technique producing seven other EfficientNet family networks, EfficientNet-B1–B7. While EfficientNet improved the current methods, further research implied EfficientNets had several setbacks, including the training slowness with very large images and depthwise convolutions in early layers. This led to the creation of the EfficientNetV2 family, which contains three different networks (S, M, L) that use similar compound scaling as presented in the earlier work, with additional optimizations [20]. In this thesis, the smallest EfficientNetV2-S network was used. The architecture of the network is shown in table 3.1.

Table 3.1. *EfficientNetV2-S architecture. Reproduced from [20].*

Stage	Operator	Stride	Channels	Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv4, k3x3, SE0.25	1	160	9
6	MBConv4, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

The following subsections give an overview of the particular operator structures used in the network.

3.2.1 Mobile inverted Bottleneck operators

Mobile inverted Bottleneck (MBConv) is the primary building block of the original EfficientNet family. The structure was originally presented in the work of Sandler et al. [27] and can be divided into four different parts: regular convolution, depthwise convolution,

Squeeze-and-Excitation layer (SE), and final convolution to reduce the size back to the original input dimensions. The structure includes a shortcut connection, as shown in Figure 3.3.

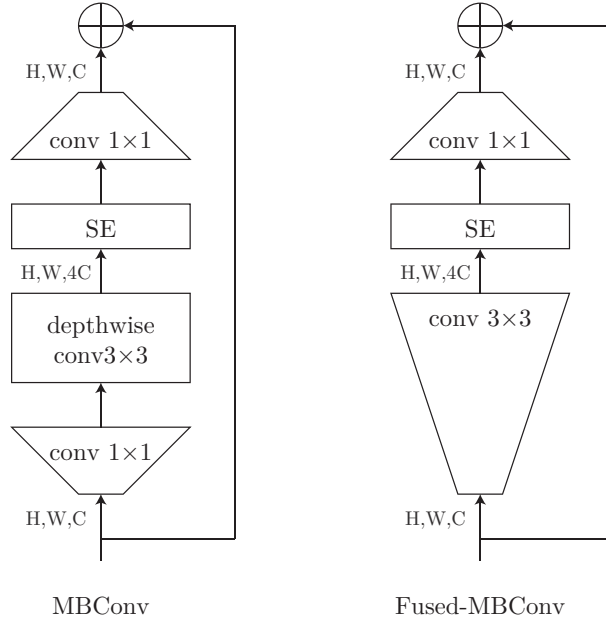


Figure 3.3. Main building blocks of EfficientNetV2: MBConv and Fused-MBConv. Reproduced from [20].

Figure 3.4 shows a more detailed example. The first convolutional layer expands the number of input channels by a chosen expand ratio. Table 3.1 mentions the expansion ratio in the block structure name, e.g., MBConv4.

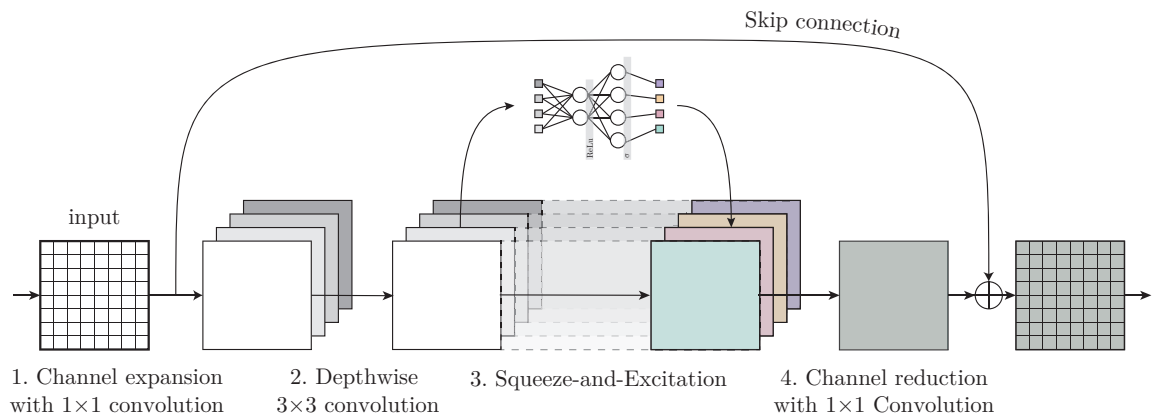


Figure 3.4. Mobile inverted Bottlenet block (MBConv).

Depthwise separable convolution block assigns a 3x3 convolution kernel to each channel, performing a spatial convolution. After passing data through SE-block, 1x1 convolution

scales channels back to the original input dimensions. Skip connection adds the original input to the result.

The improved EfficientV2 family proposed a change in the original architecture by introducing Fused Mobile inverted Bottleneck blocks (Fused-MBConv), shown in Figure 3.3. The structure was introduced by Gupta and Tan [28] to better use mobile or server accelerators. The system is similar to MBConv, but the first 1×1 convolutional expansion and 3×3 depthwise convolution are replaced with a regular 3×3 convolution layer.

3.2.2 Squeeze-and-Excitation blocks

Squeeze-and-Excitation (SE) block was introduced for CNNs in Hu et al. [29]. The block aimed to improve previous models by increasing the sensitivity to channel-wise relationships. The block is used in MBConv and Fused-MBConv structures, as shown in Figure 3.3. The concept of SE is illustrated in the Figure 3.5.

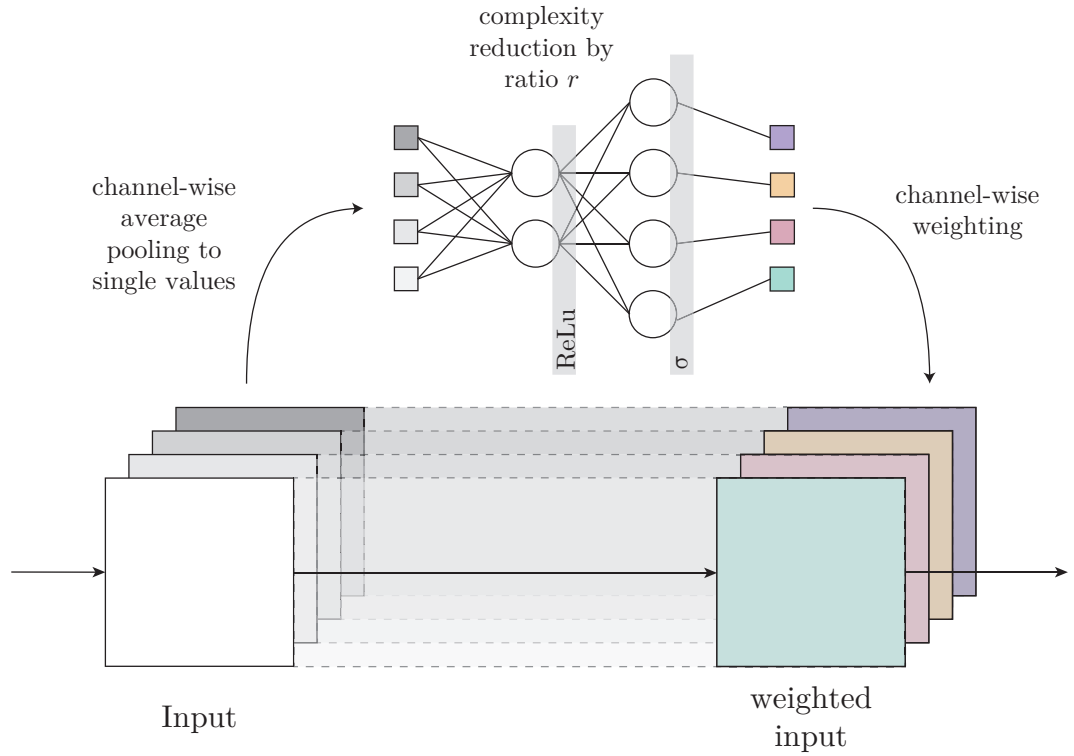


Figure 3.5. Simplified example of Squeeze-and-Excitation (SE) block. The input is squeezed into single values and passed to fully connected (FC) layers. The amount of nodes is reduced by ratio r in the first FC layer. The first layer uses ReLU as an activation function; the second sigmoid σ . The output is used to weigh the original input channel-wise.

First, the structure applies average pooling to squeeze each input channel into a single numeric value. After that, the values are passed to a fully connected (FC) layer, where

the number of neurons is scaled to channels divided by reduction ratio r . Non-linearity is added by using the ReLU activation function. The second FC layer expands the number of nodes to the original number of input channels. The result is passed through the Sigmoid activation function to give each channel a smooth gating function. Finally, the resulting values weight each feature map of the original convolutional input block. [29]

3.3 Discrete Fourier Transformation

A French mathematician and physicist, Jean Baptiste Joseph Fourier (1768–1830), is well known for his mathematical discoveries and contributions to science. In 1807 he presented a paper to the Institut de France with a controversial claim that any periodic signal could be represented as a sum of chosen sinusoidal waves. [30, p.141] The Fourier theorem is used in various fields such as communication, signal, and image processing.

Fourier transform can be divided into four categories according to the basic signal types, as illustrated in Figure 3.6. As computers can only work with discrete and finite information, discrete Fourier transformation (DFT) and its more efficient algorithmic implementation Fast Fourier Transformation (FFT), are the most applied transformations in digital signal processing.


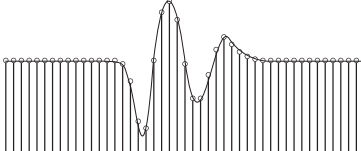

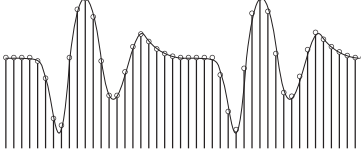
Transformation	Signal type	
Fourier Transformation		continuous aperiodic
Discrete Time Fourier Transformation		discrete aperiodic
Fourier Series		continuous periodic
Discrete Fourier Transformation		discrete periodic

Figure 3.6. Different Fourier transformations. Reproduced from [30].

The common usage of DFT can be divided into three categories; Spectral analysis, the frequency response of systems, and convolution through Frequency Domain [30, p.169].

In spectral analysis, DFT extracts information about the frequency, phase, and amplitude of the signal's component sinusoids. The signal's spectral density can reveal useful information about the signal's periodicity and other features. Using the inverse discrete Fourier transform (IDFT), the signal can be observed in its original domain without losing any information during the process.

Frequency response is a characteristic of a linear system, describing how the system changes the amplitude and phase of cosine waves passing through it. DFT can convert the system's impulse response into its frequency response, enabling further analysis of the system. Finally, the costly computation of convolution can be simplified with FFT due to the properties of the Fourier transform. The convolution of two functions in the time domain equals multiplying their respective Fourier transforms in the frequency domain. Convolution is used in digital signal processing, e.g., to design digital filters.

Discrete Fourier Transformation is defined for the function $f(x)$, $x = 0, 1, \dots, M - 1$, as

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M}, u = 0, 1, \dots, M - 1. \quad (3.1)$$

The inverse Discrete Fourier Transformation is defined

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M}, x = 0, 1, \dots, M - 1. \quad (3.2)$$

Digital images can be considered as two-dimensional discrete functions $f(x, y)$, where x and y are the spatial coordinates and the value of f is the grayscale or intensity of the pixel [24]. Discrete Fourier Transformation for a two-dimensional image plane is defined

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}, \quad \begin{matrix} u = 0, 1, \dots, M - 1, \\ v = 0, 1, \dots, N - 1. \end{matrix} \quad (3.3)$$

Similarly, inverse discrete Fourier transformation for image plane can be defined

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}, \quad \begin{matrix} x = 0, 1, \dots, M - 1, \\ y = 0, 1, \dots, N - 1. \end{matrix} \quad (3.4)$$

Color images follow the spatial representation but usually contain three channels for red, green, and blue intensities following the RGB -color model. DFT pre-processing for images is done by performing DFT on each channel separately. When DFT is applied

to three-channel RGB images, the output consists of real and imaginary parts for each channel, as shown in Figure 3.7. The real and imaginary parts describe the amplitudes of sine and cosine waves of the original signal.

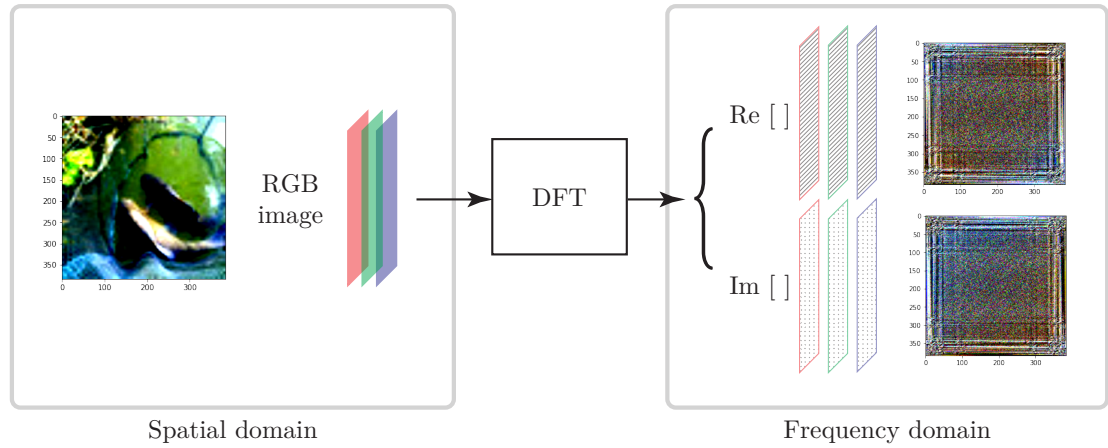


Figure 3.7. Discrete Fourier Transformation for RGB image. Sample image from ImageNet dataset [9].

4. EXPERIMENTAL SETUP

The most common training method used with neural networks falls into the category of supervised learning. In supervised learning, the training dataset provides correct responses, and the algorithm generalizes to respond correctly to inputs [31]. Training a neural network equals updating its parameters, such as layer weights and biases. A simplified training process can be divided into three stages, as illustrated in Figure 4.1.

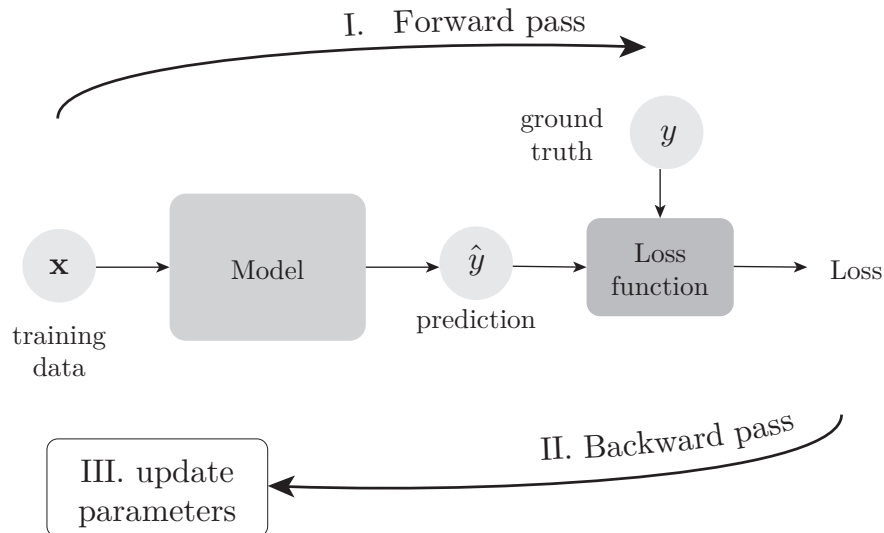


Figure 4.1. A typical supervised training process of neural networks consists of three stages.

In the forward pass, the neural network produces a prediction \hat{y} for the given input x . In image classification, the prediction corresponds to the predicted image class. A loss function compares the prediction \hat{y} with the ground truth y included in the training data, computing an error for the training input. Updating parameters is typically achieved through optimizing the loss function via a minimization algorithm and backpropagation, also called backward pass.

Successful training is a combination of a chosen network architecture, loss function, minimization algorithm, and fine-tuned hyper-parameters. Section 4.1 goes through the used test setups in detail. Section 4.2 explains the evaluation metrics that were used to compare the learning between RGB and DFT network.

4.1 Training setup

Two different neural networks were created by using Python programming language in a Google CoLab environment, using GPUs and an open-source library PyTorch [32]. The model for EfficientNetV2-S was imported from torchvision.models library. A reduced version of ImageNet for the 2012 contest [9] was used as the dataset, containing 1.2 million 64x64 sized images of 1000 classes.

An instance of the EfficientNetV2-S model with pre-trained weights was used as a baseline setup. The accuracy was computed directly for the resized ImageNet evaluation set. The first experimental setup included an instance of EfficientNetV2-S, this time training the network from scratch. Images were pre-processed to match the EfficientNetV2-S input structure.

For the second setup, the first convolutional layer of the EfficientNetV2-S instance was replaced with a convolutional layer of 6 input channels. Besides the regular image pre-processing, RGB images were passed through a 2D Fast Fourier Transformation. The image FFT resulted in 6 channels containing the real and imaginary parts of RGB channels. The parts were stacked and fed to the neural network.

Training from scratch was conducted in 10 epochs for both RGB and DFT networks. The training process was conducted using the Stochastic Gradient Descent (SGD) optimization algorithm, coupled with Cross-Entropy (CE) loss function. The learning rate was scheduled to decrease in intervals.

4.1.1 Optimizer: Stochastic Gradient Descent

The concept of partial derivatives is used to minimize the loss function. The gradient of loss contains all the partial derivatives with respect to model parameters and points to the direction where the function increases the most. By contrast, the negative gradient of loss points to the direction where the function decreases the fastest.

Backpropagation computes the gradient of loss by iterating backward, applying the chain rule to find partial derivatives with respect to each model parameter. Parameters are updated by taking a step toward the negative gradient. The process is called gradient descent (GD) and gives an update rule in Equation 4.1 for the parameters θ of time step $t + 1$

$$\theta^{t+1} = \theta^t - \mu \nabla_{\theta} \mathcal{L}(\theta^t), \quad (4.1)$$

where μ is the learning rate, $\nabla_{\theta} \mathcal{L}$ is the gradient of the loss with respect to the model parameters at time step t . The learning rate μ scales the step size towards the negative

gradient.

Stochastic Gradient Descent (SGD) is an improved version of the GD. SGD randomly chooses a single sample or batch for gradient computation and updates the parameters accordingly. Typically SGD is computed over mini-batches to improve computational efficiency. The new update rule in Equation 4.2 is formulated

$$\theta^{t+1} = \theta^t - \mu \sum_{b=1}^B \nabla_{\theta} \mathcal{L}_b(\theta^t), \quad (4.2)$$

where B stands for the batch size, $\nabla_{\theta} \mathcal{L}_b$ is the gradient of the batch loss with respect to model parameters θ .

The modern implementations of SGD include additional parameters such as momentum and weight decay. Momentum uses the statistics over the past values of the gradient to fine-tune the step size towards the gradient. Weight decay is a regularization strategy to drive the weights closer to the origin [23, p.227].

In the experiments of this thesis, SGD was used with an additional Momentum factor set to 0.9 and a weight decay of 0.0005. The optimizer was chosen by experimenting with different minimization algorithms. Momentum factor and weight decay were based on existing similar image classification setups in the field. The learning rate μ was scheduled to decrease in intervals: 0.1 for epochs 1–3, 0.01 for epochs 4–6, and 0.001 for epochs 7–10. The learning rate schedule was limited to 10 epochs due to long processing times. The schedule was adjusted during the first RGB model's training by observing the training's loss development.

4.1.2 Loss criterion: Cross Entropy

Cross Entropy (CE) measures the difference between two probability distributions for a given random variable or set of events. In the classification task, it can be interpreted as the cross-entropy between the training data and the model distribution [23]. CE loss is typically used with classification problems that contain multiple classes. The loss penalizes probabilities far from the actual expected value.

Mathematically CE can be formulated

$$\mathcal{L}_{CE} = - \sum_{i=1}^C y_i \log(p(y = i|\mathbf{x})), \quad (4.3)$$

where C is the number of classes, y is the binary indicator for truth label of i^{th} class and $\log(p_i)$ is the natural logarithm of SoftMax probability for the i^{th} class.

4.2 Evaluation Metrics

The learning was evaluated by tracking and observing losses of the training process and computing top-1, -2, and -5 accuracies for the models. Calculating Top accuracies is a commonly used method to evaluate how neural networks work in image classification tasks. Besides computing top errors, it is a leading metric in many current scientific papers of the field [20][19][33].

Top-1 is the accuracy which tells the percentage of correct predictions the model makes. The method can be extended to top-2 and top-5 accuracies, which tell the percentage of predictions where the classifier included the correct class among the top 2 and 5 predictions.

Class-wise accuracies were calculated by passing evaluation images of each class through baseline, RGB, and DFT models. The number of evaluation images per class varied from 35 to 50 images. The differences between RGB and DFT networks' class-wise accuracies were calculated to find and highlight classes that most separate the network performances. Class-wise accuracies also revealed which classes DFT network could classify better than the RGB network.

5. RESULTS AND ANALYSIS

5.1 Accuracy and loss development

Figure 5.1 presents the top accuracies of RGB and DFT networks. The baseline network with pre-trained parameters reached a top-1 accuracy of 56,0%. It is less compared to the original EfficientNetV2-S model with top-1 accuracy of 83,9% [20], but the difference could be explained by the reduced 64x64 resolution of the training data.

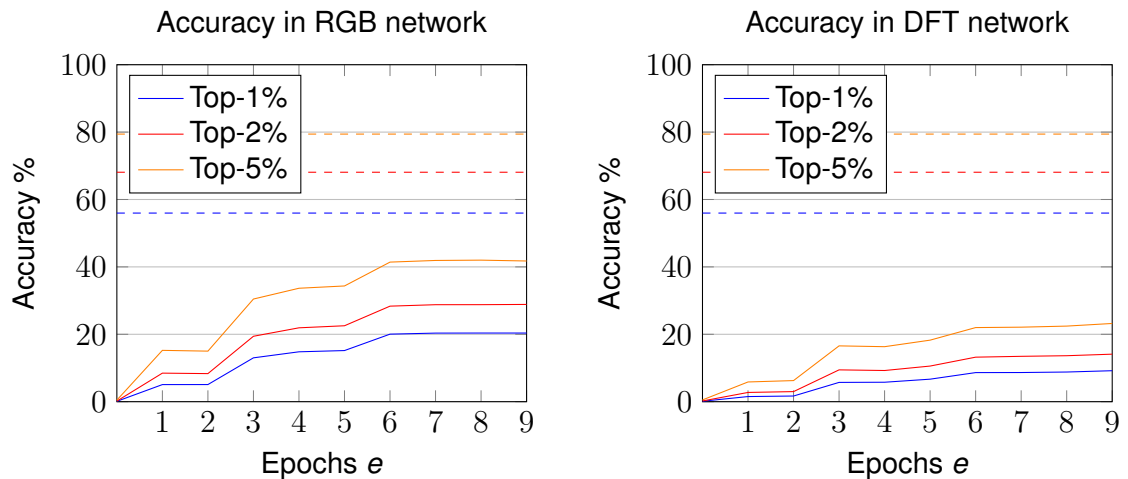


Figure 5.1. Accuracy development over trained epochs. Baseline accuracies are marked with dashed lines.

The RGB network reaches a top-1 accuracy of 20.37% after ten training epochs. The improvement between the two first epochs is slight and indicates the model has learnt the extractable features from the images. When the learning rate shifts from 0.1 to 0.01, accuracy grows steadily for the following 3 epochs. The final 4 epochs with a learning rate 0.001 bring more minor accuracy improvements.

DFT network reaches top-1 accuracy of 9,19%. The training for DFT-network has similar distinguishable phases following the learning rate schedule. The improvements are smaller when compared to RGB-network. When shifting from a learning rate of 0.1 to 0.01, the model achieves only a 4% improvement in top-1 accuracy. In contrast to the DFT model, the accuracy improvement of the RGB model in the same phase is around 8%.

Notably, the top-1 accuracies remain small in comparison to the baseline network. This could be explained by several limitations in the experimental setups. The baseline model is trained with the original ImageNet 2012 competition dataset with a higher resolution than the 64x64 images. Due to memory constraints of the environment, the training batch size was reduced to 64, while the backbone recipe worked with batch size 128. The baseline network recipe for pre-trained weights also includes over 600 training epochs. The small amount of epochs reflects in the results.

Observing the loss graph 5.2, both networks have three distinguishable learning phases following the learning rate schedule. Over ten epochs, the RGB network reaches a lower loss than the DFT network. The RGB network learns more during the initial three epochs, as seen in the accuracy graph 5.1, and the loss decreases more during the learning rate shifts compared to DFT-network.

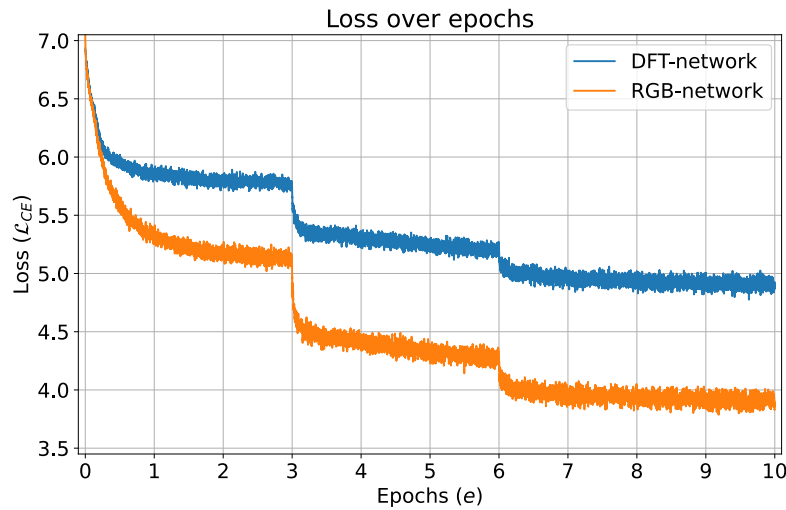


Figure 5.2. The development of loss over iterated epochs. Both RGB and DFT networks have distinguishable learning phases. The changes in the learning rate schedule can be seen at the start of epochs 3 and 6.

Following both loss and accuracy curves in Figures 5.1 and 5.2, the model could benefit from further training epochs, as the values are not entirely saturated.

5.2 Class-wise accuracy

Figure 5.3 shows the five classes that caused the most difference between RGB and DFT network accuracy. The baseline accuracy of these classes gained 77–100% top-1 accuracy, which is higher than the overall top-1 accuracy of 56.0%. RGB network follows the same trend with accuracies of 53–74%, above the average accuracy of 20.37%. DFT network has difficulties with classes African grey and barracouta; their top-1 accuracies remain below the average of 9,19%. Leaving out the failed classifications with 0% top-1

accuracy, the second graph in Figure 5.3 shows the five classes with the least difference in performance.

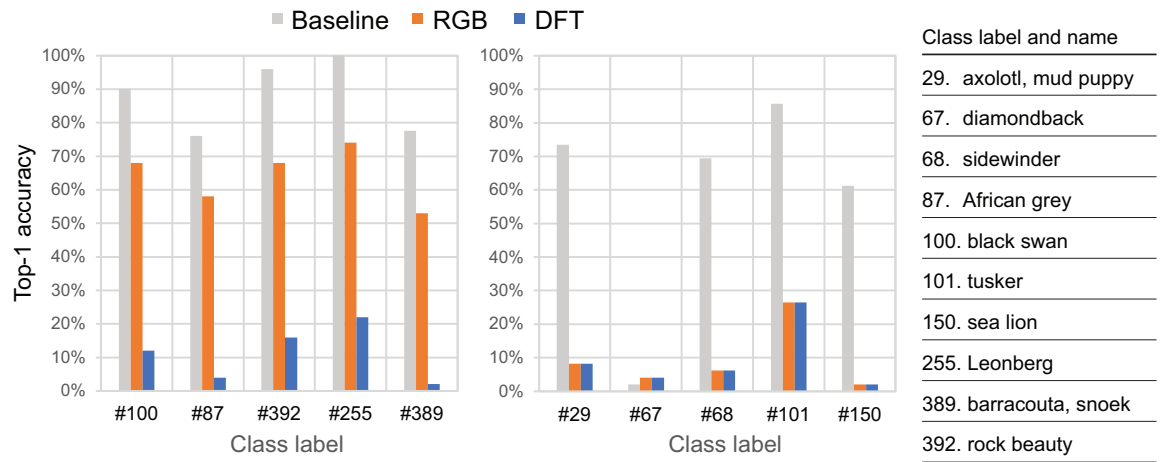


Figure 5.3. Top-1 accuracies for classes with highest and lowest accuracy difference between RGB and DFT network.

When five classes with the least accuracy difference in RGB and DFT networks were looked for, it was revealed that both RGB and DFT networks failed in classifying 61 classes. If the metrics are widened to top-2 accuracy allowing a second guess, networks fail in 25 classes. The most challenging classes for both networks are 502., 740., 813. and 876. shown in Figure 5.4. Neither of the networks could classify evaluation set images into top-5 predictions.

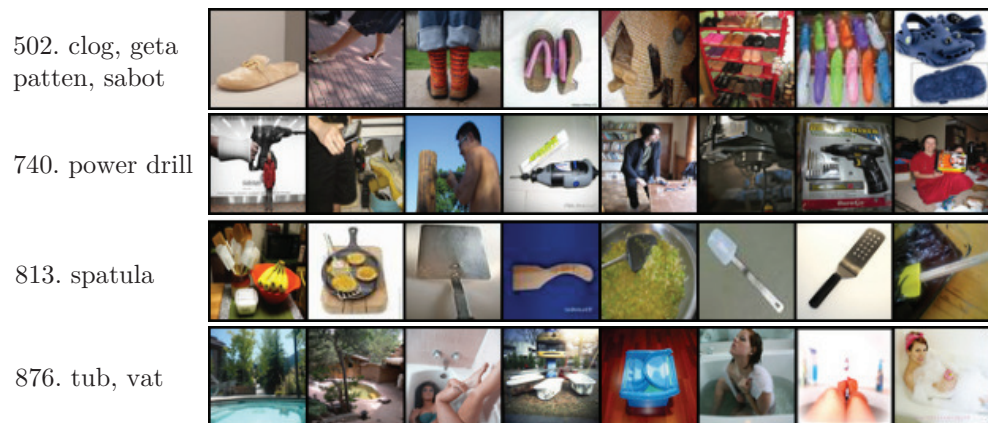


Figure 5.4. Classes which both DFT and RGB network failed to identify. The evaluation set images were incorrectly classified, and the correct label was not included in the top-5 predictions. Sample images from ImageNet [9].

Most unidentified classes are devices, everyday items, or different dog breeds. While animals always appear in distinct shapes and colors, the form of an everyday object can have more variance. For example, the visual definition of a spatula class is broader than

that of a zebra class. On the other hand, ImageNet contains 120 different dog breeds. As all the dog breeds have common features, it can be hypothesized that networks mix up different dog breeds easily. Overall, the results indicate networks would require more training data to learn to classify the most difficult classes.

Some classes performed considerably better in the DFT network, the top five shown in Figure 5.5. Visually, there is not much in common between the classes. Notably, the DFT network reaches much higher top-1 accuracy with the classes than the DFT network's overall average accuracy of 9.19%. The DFT accuracy with the white shark class is the same as the baseline accuracy. Similarly, the DFT accuracy with the koala class competes with baseline accuracy with only a 10% difference.

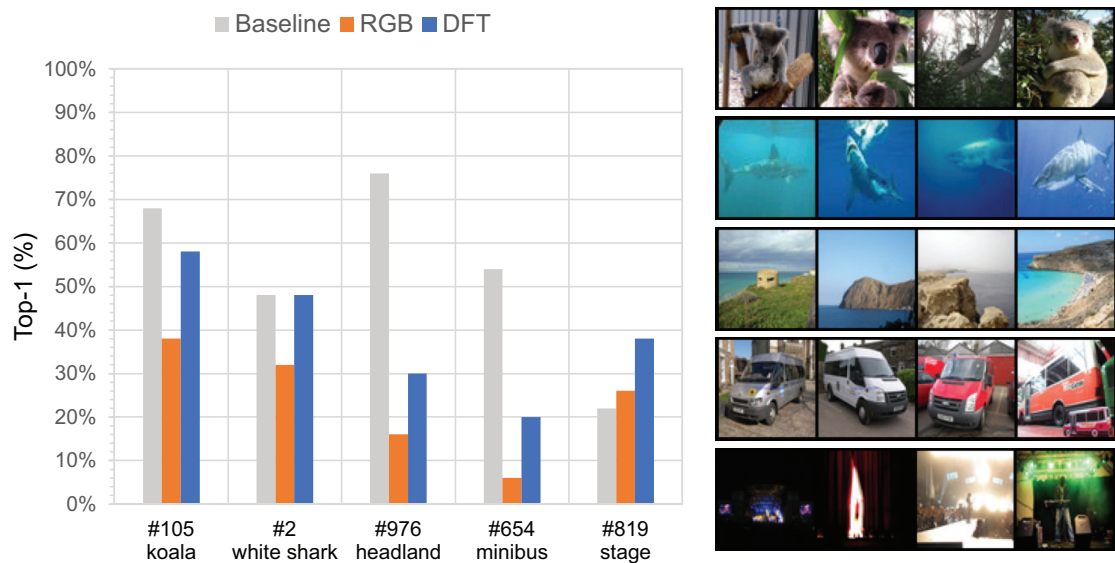


Figure 5.5. Classes which DFT network classified better than RGB network. Sample images from ImageNet [9].

The results show a lot of variance between the class-wise accuracy performances. Only a few classes are more easily classified with the DFT than the RGB network. Although the class-wise accuracies show higher results than the overall top1-accuracies of each network, it should be noted that the evaluation set only includes 35–50 images per class.

6. CONCLUSION

This thesis aimed to study how DFT pre-processing images affect the deep learning process of convolutional neural networks in image classification tasks. Neural network setups were created for RGB and DFT pre-processed images, using EfficientNetV2-S as the basic model. The learning was evaluated by observing the top accuracies of the networks over ten training epochs. The class-wise top-1 accuracies were computed to find and highlight classes separating the RGB and DFT performances the most. Results were compared to the pre-trained baseline model of the same backbone network.

The results suggest that the CNN-based backbone was able to learn from both Fourier-transformed and spatial RGB data. However, the spatial image model reached better accuracy than the DFT network over the trained epochs. The trained ten epochs produced modest results in comparison to the baseline model. Higher accuracy could be reached by extending the training over more epochs and fine-tuning the model's hyper-parameters. Furthermore, the chosen backbone included batch normalization layers. It could be beneficial to examine the effect of normalization in DFT-network by, e.g., switching off the normalization layers or designing DFT-specific normalization layers. Studying normalization could play a key role in improving DFT-network results.

It is worth noting that the PyTorch implementation of EfficientNetV2-S was not optimized for a 64x64 input resolution. Easily accessible smaller datasets, such as CIFAR-10, could be used for faster learning with more epochs. Faster learning enables a more flexible test environment for fine-tuning the model hyper-parameters, allowing the possibility to experiment in a more versatile manner.

Further studies could determine what kind of inputs benefit DFT pre-processing the most. The class-wise top accuracies indicate there could be a place for studying different DFT-discoverable patterns. Whether DFT can prove its potential in image classification tasks remains to be seen. As DFT is often used to filter noise from images, research on DFT in neural networks that specialize in noise reduction could be suggested.

Within the scope of this thesis, the training process was long enough to prove that effective features can be learnt from DFT images. This thesis focused on DFT as a pre-processing method, but the technique could be experimented with further as an internal part of neural network architectures. The internal structures could benefit from the frequency domain

presentation and exploit Fourier Transformation properties.

REFERENCES

- [1] E. A. M. van Dis, J. Bollen, W. Zuidema, R. van Rooij, and C. L. Bockting, “ChatGPT: Five priorities for research”, eng, *Nature (London)*, vol. 614, no. 7947, pp. 224–226, 2023, ISSN: 0028-0836.
- [2] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, “A survey of deep learning techniques for autonomous driving”, *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.
- [3] S. Suratkar, S. Bhiungade, J. Pitale, K. Soni, T. Badgujar, and F. Kazi, “Deep-fake video detection approaches using convolutional–recurrent neural networks”, eng, *Journal of control and decision*, pp. 1–17, 2022, ISSN: 2330-7706.
- [4] MOT Oxford Dictionary of English, *Deep learning*. [Online]. Available: https://www.sanakirja.fi/oxford_english/english-english/deep%5C%20learning (visited on 03/15/2023).
- [5] Scopus, *Abstract and citation database of peer-reviewed literature – scientific journals, books and conference proceedings*. [Online]. Available: <http://www.scopus.com/home.url>.
- [6] H. Touvron, A. Vedaldi, M. Douze, and H. Jegou, “Fixing the train-test resolution discrepancy”, in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/d03a857a23b5285736c4d55e0bb067c8-Paper.pdf.
- [7] O. Rippel, J. Snoek, and R. P. Adams, “Spectral representations for convolutional neural networks”, *Advances in neural information processing systems*, vol. 28, 2015.
- [8] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, “Deep learning for audio signal processing”, *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, May 2019, ISSN: 1941-0484. DOI: 10.1109/JSTSP.2019.2908700.
- [9] O. Russakovsky, J. Deng, H. Su, *et al.*, “ImageNet Large Scale Visual Recognition Challenge”, *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.
- [10] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity”, *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943, ISSN: 1522-9602. DOI: 10.1007/BF02478259.

- [11] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.", *Psychological review*, vol. 65, no. 6, p. 386, 1958.
- [12] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms", Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
- [14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [16] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.
- [17] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning", *arXiv preprint arXiv:1611.01578*, 2016.
- [18] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 104–12 113.
- [19] X. Chen, C. Liang, D. Huang, *et al.*, "Symbolic discovery of optimization algorithms", *arXiv preprint arXiv:2302.06675*, 2023.
- [20] M. Tan and Q. Le, "EfficientNetV2: Smaller Models and Faster Training", in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 10 096–10 106. [Online]. Available: <https://proceedings.mlr.press/v139/tan21a.html>.
- [21] W. Ye and H. Chen, "Human activity classification based on micro-doppler signatures by multiscale and multitask fourier convolutional neural network", *IEEE Sensors Journal*, vol. 20, no. 10, pp. 5473–5479, May 2020, ISSN: 1558-1748. DOI: 10.1109/JSEN.2020.2971626.
- [22] O. Elharrouss, Y. Akbari, N. Almaadeed, and S. Al-Maadeed, *Backbones-review: Feature extraction networks for deep learning and deep reinforcement learning approaches*, 2022. arXiv: 2206.08016 [cs.CV].
- [23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [24] R. C. Gonzalez, *Digital image processing*, eng, Fourth edition, Global edition. Harlow, Essex, England: Pearson Education Limited, 2018 - 2018, ISBN: 9781292223049.

- [25] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks”, in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, PMLR, Sep. 2019, pp. 6105–6114. [Online]. Available: <https://proceedings.mlr.press/v97/tan19a.html>.
- [26] M. Tan, B. Chen, R. Pang, *et al.*, “MnasNET: Platform-aware neural architecture search for mobile”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2820–2828.
- [27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, *MobileNetV2: Inverted Residuals and Linear Bottlenecks*, 2019. arXiv: 1801.04381 [cs.CV].
- [28] S. Gupta and M. Tan, *EfficientNet-EdgeTPU: Creating Accelerator-Optimized Neural Networks with AutoML*, Accessed on 14.04.2023, Aug. 2019. [Online]. Available: <https://ai.googleblog.com/2019/08/efficientnet-edgetpu-creating.html>.
- [29] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, *Squeeze-and-excitation networks*, 2019. arXiv: 1709.01507 [cs.CV].
- [30] S. Smith, *Digital signal processing: a practical guide for engineers and scientists*. Elsevier, 2013.
- [31] S. Marsland, *Machine Learning: An Algorithmic Perspective* (Chapman & Hall CRC machine learning & pattern recognition series), eng, 2nd ed. CRC Press, 2015, ISBN: 1466583282.
- [32] A. Paszke, S. Gross, S. Chintala, *et al.*, “Automatic differentiation in pytorch”, in *NIPS-W*, 2017.
- [33] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, *Coca: Contrastive captioners are image-text foundation models*, 2022. arXiv: 2205.01917 [cs.CV].