Universiteit Leiden
The Netherlands

Opleiding Informatica

Feature Extraction Methods
as Backbone in Image Forgery Detection

Sanne Eeckhout

Supervisors:
Prof. dr. Michael S. Lew & Dr. Erwin M. Bakker

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

May 2023

Acknowledgements

I want to thank Prof. Dr. Michael Lew for his support and supervision during my thesis, as well as my second supervisor Prof. Dr. Erwin Bakker. I also want to thank my family and friends for their support. Due to the rise of AI in academics, I want to provide a clear description of any AI assistants that were used in the process of this thesis, with the intent of inpiring others to do the same. For this thesis, Grammarly was used to check for any grammar mistakes and readability issues, and ChatGPT was used to facilitate synonyms for often used words. Thank you for reading this thesis.

Abstract

Images play an essential role in our digital age, which has given rise to the issue of image forgery. Fortunately, researchers are actively working to develop methods to detect image forgery. This paper examines the impact of different feature extraction methods as backbone networks for algorithms that detect image forgery and image manipulation. The feature extraction methods employed for this study, VGG16, InceptionResNetV2, EfficientNetB4, and XceptionNet, were carefully selected with a discerning balance between inference speed and accuracy. Incorporating the four different feature extraction methods on a robust dataset, developed specifically for this study, into a single image forgery detection network FDFtNet, this study demonstrates that better performing feature extraction methods can improve the accuracy of existing image forgery detection methods. The results can be attributed to the increased depth and complexity compared to the older feature extraction methods such as VGG16, which is unable to capture fine details in images, a property particularly important where subtle differences may be indicative of tampering. The novel dataset ensures that the obtained results are not limited to a specific dataset but rather generalize well across various scenarios. This novel dataset significantly contributes to the field by increasing the diversity of the data and it generates new combinations of objects and scenes that were not present in existing datasets. The networks are evaluated using two novel metrics, the Detection Improvement Ratio (DIR) and the Image Forgery Efficiency Ratio (IFER), which advance our understanding of the importance of feature extraction methods in image forgery detection and provide valuable insights for researchers in this domain.

Keywords: image forgery detection, image manipulation, feature extraction, image tampering, FDFtNet, DIR, IFER

Contents

1 Introduction

In this digital age, images play an essential role in communication, entertainment, and many other domains. Unfortunately, the widespread use of digital images has given rise to image forgery, an important issue. Image forgery refers to the act of manipulating images with malicious intent. With the ease of image manipulation, it has become increasingly difficult to distinguish real images from fake ones. This leads to a rise in different types of image forgery, such as splicing, copy-move, removal, and several facial forgeries, with the most widely known methods of Deepfakes among these. This has severe implications in society, from spreading false information to causing reputational damage and potential legal issues. Therefore, image forgery detection has become a crucial area of research in computer vision and image processing, and the number of publications on image forgery detection has increased exponentially [43]. The detection of image forgery is crucial to ensure that these digital images accurately represent reality.

The detection of image forgery can be accomplished using various algorithms that employ feature extraction methods. In recent years, several feature extraction models have been designed, such as VGG16 [36], XceptionNet [5], EfficientNetB4 [39], and InceptionResNetv2 [37], which are widely used in image classification and object detection. These models have shown exceptional performance in image classification.

Despite the progress made in the field of image forgery detection, the detection of forgeries remains challenging and image forgery localization has proven to be one of the most difficult areas in digital image forensics [13]. Detecting forgeries is complicated as forgers often use sophisticated techniques to conceal their activities and as supervised learning is required, datasets that contain ground truth are crucial. Unfortunately, these datasets can be rare and as the techniques to forge images become more complex, the detection of forgeries becomes increasingly difficult.

The primary objective of this thesis is to study whether newer and higher performing feature extraction models can improve the performance of existing image forgery detection algorithms. Particularily, this study focuses on the image forgery detection algorithm FDFtNet, a state-of-the-art algorithm that contains feature extraction by XceptionNet. We study four different feature extraction models using FDFtNet, including VGG16 as a baseline model, on a novel dataset and compare their performance using two unique metrics.

The contributions made in this thesis are fourfold. Firstly, we present an extensive analysis of the impact of using different feature extraction methods as backbone networks on the performance of image forgery detection algorithms Secondly, this study presents insights into the advantages of utilizing advanced feature extraction models to detect image forgeries with the use of a novel metric called Relative Detection Improvement Ratio (RDIR). The thesis also introduces an independent novel dataset that combines data from five different datasets, and therefore contains many forgery types. This enables a reliable and robust experiment, as it ensures that the training of the algorithms is not compromised by overfitted data or focussed on one specific forgery type. Lastly, it includes an evaluation and comparison of the networks mentioned on a novel metric called the Image Forgery Efficiency Ratio (IFER), which combines the classification time, computational cost, and accuracy of each network. This sheds light on the feasibility of the practical use of the proposed image forgery

detection algorithms in media platforms.

1.1 Thesis overview

The remainder of this thesis is organized as follows. Section 2 includes the definitions that are used in this paper. In Section 3, we provide an overview of the literature on image forgery detection and feature extraction models. We also discuss the different datasets that have been proposed for image forgery detection and have been used in said networks. In Section 4 we provide a detailed analysis of the chosen image forgery detection algorithm FDFtNet, as well as an in-depth comparison of the feature extraction methods VGG16, XceptionNet, EfficientNetB4, and InceptionResNetv2. Furthermore, it involves a comprehensive overview of the datasets used to create the novel dataset for training and evaluating the networks, including a detailed description of the images in the dataset. Section 5 describes the methodology used in our experiments, including the specific techniques of pre-processing, and different parameter settings. It also includes the different evaluation metrics used, as well as the novel metrics Detection Improvement Ratio (DIR) and Image Forgery Efficiency Ratio (IFER). In Section 6, we present the results of our experiments in various figures and tables and Section 7 analyses the performance of the four feature extraction models. We compare the performance of the four models against the baseline model VGG16, and examine the factors that contribute to their performance. We also perform a detailed analysis of the results of the DIR and IFER metrics. Finally, in Section 8, we conclude the thesis by summarizing the findings and discussing future research directions in the field of image forgery detection.

This thesis is part of the bachelor program in Computer Science with a specialization in Artificial Intelligence at the Leiden Institute of Advanced Computer Science (LIACS) at Leiden University and was written under the supervision of Prof. Dr. Michael Lew and Prof. Dr. Erwin Bakker.

2 Definitions

2.1 Acronyms

IFDL	Image Forgery Detection and Localization
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
CMFD	Copy-move forgery detection

2.2 Terminology

Image forgery	Image manipulations that produce fake graphic content which falsifies some facts
Image tampering	A special type of image forgery that alters a part or multiple parts of the graphic content of a given digital image
Image manipulation	Computing techniques that paint or edit a digital image

3 Related Work

Many researchers have attempted to tackle the problem of image forgery detection and numerous datasets and feature extraction methods have been proposed in the last few years to help forgery detection methods. These detection methods can be categorized into active and passive forgery-types, and further sub-categorized into dependency. A graph illustrating these categories can be seen in Figure 1.

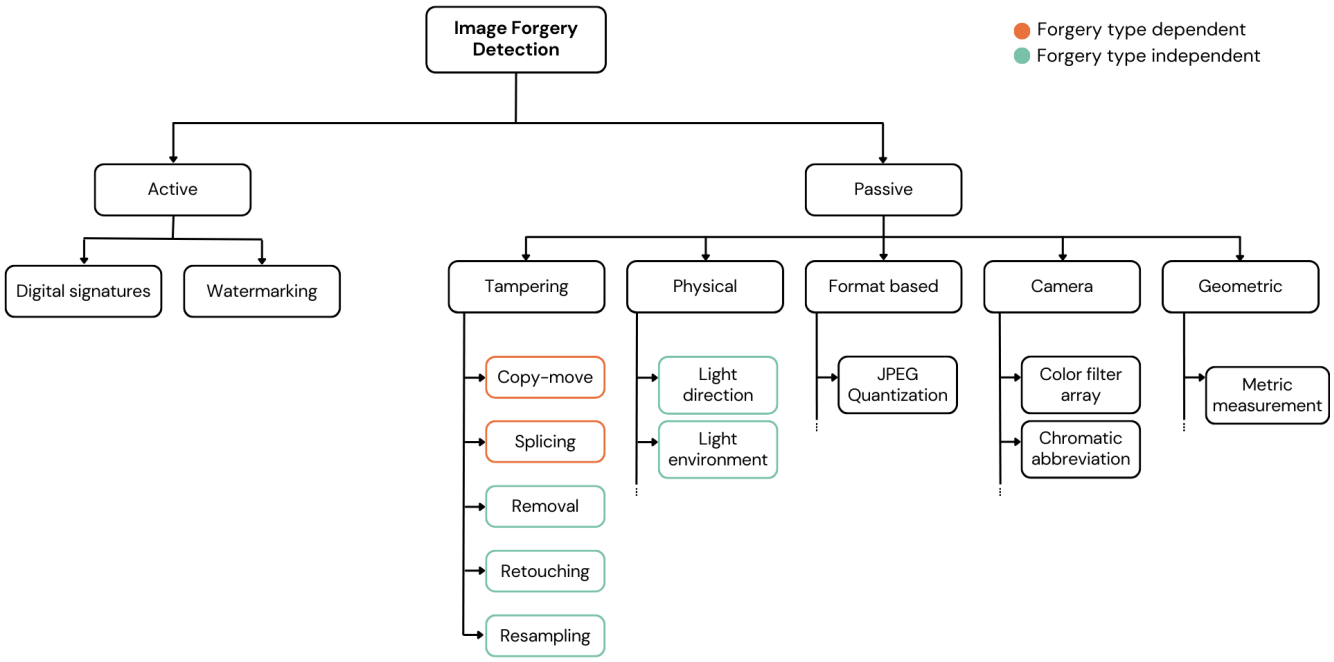


Figure 1: The different image forgery types in the field of IFDL

3.1 Networks

The current literature on image forgery detection highlights several deep learning-based approaches for the task of image forgery detection. Among them, ManTra-Net, MesoNet, FDFtNet, SPAN, and RGB-N are notable examples. Aside from these networks being categorized in those shown in Figure 1, the image forgery detectors can be categorized into their form of classification, either a binary classification of real or fake, often determined by a percentage or by predicting a localization of the tampered region by producing a mask. An example of such a mask can be found in Figure 2.

ManTra-Net [47] is a manipulation tracing network designed for detecting and localizing image forgeries with anomalous features. It was proposed by Wu et al. in 2019 and utilizes a novel manipulation tracing module to capture the traces of different image manipulation operations. It is based on a VGG16 architecture with a feature pyramid network (FPN) for multi-scale feature

Figure 2: Example of binary mask produced by a CNN-based model [3]

extraction. ManTra-Net takes an input image and produces a heatmap indicating the regions of the image that are anomalous, i.e., likely to be manipulated.

MesoNet, introduced by Afchar et al. in 2018 [1], is a compact facial video forgery detection DCNN. The network extracts compact and discriminative features from facial regions using a multi-scale aggregation of deep features and exploits the mesoscopic properties of face images. It has a total of 26 convolutional layers and uses a bottleneck design for parameter efficiency.

FDFtNet, proposed by Jeon et al. in 2020 [22], is a fake detection fine-tuning network. This network is trained using a two-stage fine-tuning approach, where the first stage involves pre-training the network on a large-scale dataset using a backbone network and the second stage involves fine-tuning the pre-trained network on a smaller target dataset to improve its detection accuracy on fake images. The FDFtNet is designed to detect and localize fake images, and it leverages the distributional difference between real and fake images to improve accuracy.

The spatial pyramid attention network (SPAN), a deep learning-based network proposed by Hu et al. in 2020 [18], is designed for image manipulation localization. SPAN uses ResNet as a backbone for feature extraction and combines a spatial pyramid pooling module with an attention mechanism to detect and localize image manipulations.

RGB-N [50] uses a multi-scale CNN architecture to learn rich features for detecting image manipulations and was introduced by Zhou et al. in 2018. The characteristic that distinguishes itself from other networks is its use of a novel RGB-normalization operation. This enables the network to capture some of the subtle differences that are between manipulated and real images.

Other notable networks in the field of image forgery detection include BusterNet, which specialized in CMFD [46], ObjectFormer, displaying the artifact of the forged feature [44], MFCN specialized in splicing forgeries [34] and PSCC-Net [26], build on the feature extraction method HRNet.

3.2 Feature extraction methods

Many different networks described in the previous paragraph use a known feature extraction model as an effective backbone to their network. An overview of these networks and their corresponding base models can be found in Table 1. The topic of feature extraction methods has been extensively

Authors	Network	Backbone	Dataset
Wu et al.	ManTra-Net [47]	VGG-16 and VGG-19	Dresden, KCMI
Hu et al.	SPAN [18]	VGG	Dresden, KCMI
Afchar et al.	MesoNet [1]	-	FaceForensics++
Jeon et al.	FDFtNet [22]	Xception, ResNet, ShallowNet and SqueezeNet	FaceForensics++, CelebA
Wu et al.	BusterNet [46]	VGG	MIT SUN, MSCOCO
Zhou et al.	RGB-N [50]	Faster R-CNN → ResNet	MSCOCO
Wang et al.	ObjectFormer [44]	EfficientNetB4	
Liu et al.	PSCC-Net [26]	HRNet	Dresden, KCMI, MSCOCO
Dong et al.	MVSS-Net [7]	FCN → ResNet	CASIA
Zhou et al.	GSRNet [49]	VGG16	CASIA, MSCOCO
Akbari et al.	PRNU-Net [2]	MISLNet	
Guo et al/	AMTEN [12]	DenseNet	CelebA

Table 1. Overview of SOTA image forgery detection methods with their feature extraction backbone models and the datasets used for training and testing

studied in scientific research. SIFT (Scale-Invariant Feature Transform) [28], was introduced in 1999 and is a feature extraction algorithm that is invariant to scale, orientation, and affine distortion. It identifies the key points in an image, which are then described by a set of descriptors. The descriptors are generated by considering the gradients of the pixels in a local region around the key point. SURF [4], created as an improvement to SIFT in 2006, uses a set of Haar wavelet responses instead of gradient information to generate the descriptors and a box filter approximation to Gaussian smoothing. However, newer methods, such as CNN-based feature extraction, have surpassed these networks in terms of accuracy and performance.

Convolutional neural networks can also be used for this purpose and have shown a lot of promise in addressing these problems, as CNNs are capable of extracting high-level features. One widely used model is VGG [36], a DCNN introduced in 2014 by Simonyan and Zisserman. VGG16 and VGG19 have fixed architectures consisting of convolutional and fully connected layers, of 16 and 19 layers, respectively.
In 2015, the Residual Network ResNet [14] was introduced as a solution to the problem of vanishing gradients in DCNNs. This model introduced residual connections between the layers to enable better gradient flow through the network. ResNet50, with 50 layers and 25 million parameters, is the most commonly used ResNet model for image recognition tasks. In 2016, the ResNetv2 [15] model was proposed as an improvement to ResNet, with better skip connections that bypass several layers of the network to enhance identity mapping. The ResNetv2 has the same number of layers and parameters as ResNet but with a different architecture.

In the pursuit of striking the perfect balance between network depth and width, the InceptionNet model was devised [38]. Its primary objective was to gfinding an optimal trade-off between the depth and width of the network, while maintaining a manageable number of parameters. To accomplish this, the architecture comprises a series of Inception modules, distinguished convolutional blocks that possess the unique ability to efficiently capture information at various spatial scales.

Building upon the foundations of InceptionNet, XceptionNet [5] was proposed. Born from the concept of an extreme version of InceptionNet, it embraced the utilization of depthwise separable

convolutions and pointwise convolutions, strategically woven together to construct an extreme version of InceptionNet.

As algorithms to detect image forgery are improving, feature extraction methods are improving as well. A newer feature extraction method, EfficientNet [39], is a convolutional neural network that uses a technique called compound scaling method and has obtained state-of-the-art performance on the ImageNet and the CIFAR-100 datasets. The network contains several different versions, scaling the baseline network up from EfficientNet-B0 to EfficientNet-B7.
The InceptionResNet [37] model was introduced as a combination of the Inception and ResNet architectures, utilizing residual connections between inception modules to enhance the performance of the previous Inception and ResNet models.

Other notable feature extraction methods that have been utilized in image forgery detection include ShallowNet [40], SqueezeNet [20], DenseNet [19], NASNet [51], AmoebaNet [32], and ShuffleNet [48].

3.3 Datasets

State-of-the-art algorithms in image forgery detection rely heavily on the availability of appropriate datasets, as supervised learning approaches are required to accurately detect images that have been tampered with. Specifically, these approaches require training on datasets that contain ground truth labels. While these datasets are essential for advancing the state-of-the-art in image forgery detection, they are often scarce and difficult to obtain, since forged images are typically created to deceive people and may lack ground truth labels. Therefore, the majority of algorithms in image forgery detection create a synthetic dataset using manipulation techniques to train the networks. Some commonly used datasets are the DRESDEN image database [10] used by [47][26][18], MS COCO [25] seen in [46][50][49], the KCMI Kaggle Camera Model Identification [16] dataset used by [47][26][18] and the CelebA database, containing facial images of celebrities [27] used by [22] [12].

Nonetheless, these datasets are crucial for developing image forgery detection methods and therefore, several datasets were proposed in the past decade, which I will present in this section.

Rossler et al. produced the FaceForensics++ dataset [33], a dataset containing over 1.8 million images and videos of manipulations created with five widely used facial manipulation techniques. The FaceForensics++ dataset has created a benchmark for facial manipulation detection to standardize the evaluation of detection methods[1]. The FaceForensiscs++ dataset was used by [1] [22], amongst others. Another widely used benchmark dataset for image forgery detection is the CASIA Image Tampering Detection Evaluation Database [8], available in two versions, CASIAv1.0 and CASIAv2.0. CASIAv1.0 contains around 5000 images with copy-move and splicing forgeries. CASIAv2.0 contains a larger collection of tampered and authentic images, with a total of 10,000 images. The tampered images include various types of forgeries, such as copy-move, splicing, and retouching. Both CASIAv1.0 and CASIAv2.0 include the location of tampering and method of forgery in the ground truth labels and the datasets were used to test several algorithms [47][46][44][26][34][50][49][7].

[1] kaldir.vc.in.tum.de/faceforensics_benchmark

Furthermore, the DEFACTO dataset is a dataset that specifically uses splicing, copy-move, removal, and morphing from the MSCOCO dataset to automatically generate 149000 forged images, and was created in 2019 [29]. The MSCOCO dataset is a large-scale dataset with a total of 2.5 million non-forged labeled instances in 328k images produced by Microsoft [25].

In addition to these larger datasets, there are smaller datasets such as the NIST16 dataset [11] that was part of the NIMBLE Challenge 2017 evaluation and used by [47][44][26][34][50][49][7]. It contains about 100 images with a high resolution and their location of tampering as a binary mask. The COVERAGE dataset [45] contains a collection of 1000 authentic and 1000 tampered images created by copy-move forgeries and was used by [47][44][26][34][50][18][7]. In this dataset, the ground truth incorporates information about the type and location of tampering. The COLOMBIA Image Splicing Detection Evaluation Dataset [31] used by [47][44][26][34][50][7][18] contains a collection of 120 authentic and 120 spliced images primarily created by splicing and its ground truth comprises information about the location of the splicing and whether the image is authentic or has been spliced.

4 Methods

4.1 FDFtNet

FDFtNet, the Fake Detection Fine-tuning Network, was designed for the robust detection of fake images. It integrates several advanced techniques to enhance the performance and accuracy of their detection algorithm. The architecture, depicted in Figure 3 combines a Fine-Tune Transformer (FTT) with a pre-trained CNN as a backbone, and the MobileNet block (MBblockV3) to provide efficient feature extraction via different convolution and structure techniques. By leveraging well-established CNN architectures for fake image detection, FDFtNet offers a reusable fine-tuning network that enhances the performance of existing backbone CNN models, which were not specifically designed for detecting fake images. FDFtNet was originally tested on four backbone CNN models, SqueezeNet, ShallowNetV3, ResNetV2, and XceptionNet, where the latter yielded the best performance. Due to its clear use of a backbone CNN model, efficiency, and availability, FDFtNet was chosen as the primary network for this study to compare various CNN models as backbone networks. This section will further focus on FDFtNet and its architecture.

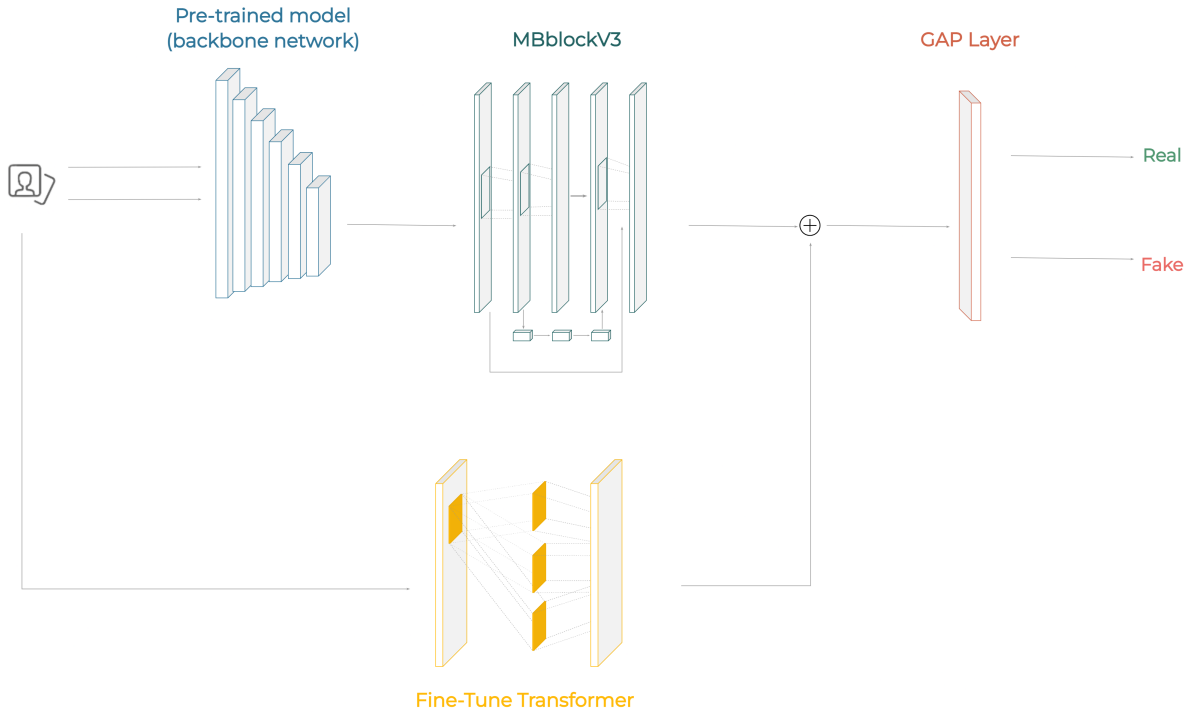


Figure 3: The architecture of FDFtNet [22]

To enhance the model's robustness and its ability to overcome the challenge of limited fine-tuning datasets, data augmentation is applied via the Cutout method [6]. This technique helps to address the problem of overfitting, where the model performs well on the training data but poorly on new, unseen data. By generating new training samples through applying squared zero masks, Cutout can increase the diversity and quantity of data available for fine-tuning, which proved to increase the model's performance and generalization ability in the experiments of FDFtNet [22].

The FTT employs self-attention modules, as illustrated in Figure 4, to extract distinct features from images. The input image is divided into three feature spaces $f(x)$, $g(x)$, and $h(x)$, which are obtained via a 1x1 convolution filter. $M = 3$ iterations are applied to the input images, which was empirically set [22].

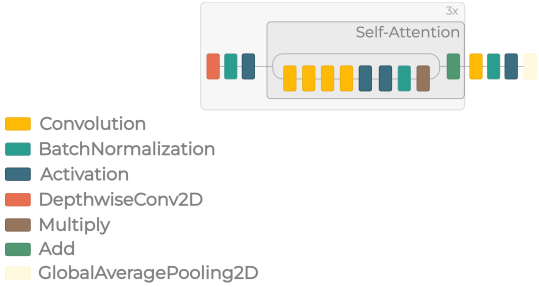


Figure 4: The architecture of the Fine-Tune Transformer of FDFtNet[22]

In addition to FTT, FDFtNet also incorporates the MobileNet Block, also called the MBConvBlock or MBblockv3, to enhance the model's feature extraction capabilities and to explore the image feature space through inverted residual structure and a linear bottleneck. The MBblockV3 is added to the pre-trained model after removing the classification layers, enabling the network to extract features using different convolution and structure techniques. MBblockV3 utilizes depthwise separable convolutions and was chosen due to its computational efficiency and accurate feature extraction on a pre-trained feature space. By combining the FTT with the MBblockV3, FDFtNet can extract more complex and diverse features from images, improving the model's accuracy and robustness. After this, the Squeeze-and-Excitation (SE) blocks [17] are applied in the bottleneck layer, to improve the representational power of the network by enabling it to perform dynamic channel-wise feature recalibration. The MBblockV3 is in turn repeated $N = 3$ times, also determined empirically. Details of the MBblockV3 can be found in Figure 5.

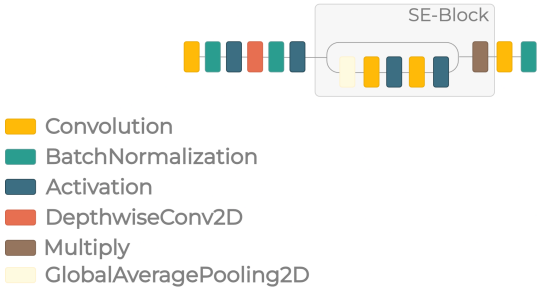


Figure 5: The architecture of the MBConv Block (or MBblock) as in MobileNet [35], used in FDFtNet

4.2 Backbone architectures

Keras documented an extensive overview of their implemented models and their top-1- and top-5 accuracy[2], which refers to the model's performance on the ImageNet validation dataset. The stated

[2]Keras API, `https://keras.io/api/applications/`, accessed 02.2023

feature extraction methods and their keras documentation can be seen in Table 3. These models can be used as backbone models, meaning they serve as a feature extraction model before passing the weights to a model that predicts the binary classification or creates a mask of the forged region.

Feature Extraction Methods	Top-1 Accuracy	Top-5 Accuracy	Time (ms) per inference step
VGG16	71.3%	90.1%	4.2
XceptionNet	79.0%	94.5%	8.1
EfficientNetB0	77.1%	93.3%	4.9
EfficientNetB4	82.9%	96.4%	15.1
InceptionResNetv2	80.3%	95.3%	10.0

Table 2. The performance of the feature extraction methods VGG16, XceptionNet, EfficientNetB0, EfficientNetB4, InceptionResNetV2 and their corresponding times per inference step

A majority of image forgery detection models that have demonstrated notable performance have utilized a VGG16 architecture, thereby I've designated it as a baseline model for this study. To accurately compare my results to that of the original FDFtNet paper, their used feature extraction method XceptionNet was included as a newer baseline compared to VGG16. To achieve expedited classification while maintaining superior performance in comparison to both VGG16 and XceptionNet, the subsequent models were selected with a discerning balance between inference speed and accuracy. In this section, the architectures of the two chosen feature extraction methods, EfficientNetB4 and InceptionResNetV2 are shown exactly as they are implemented in Section 5 and their design choices are described in great detail, with characteristics and limitations on each architecture. For comparison, the VGG16 and XceptionNet architectures are described as well.

4.2.1 VGG16

The VGG16 model has been a popular network for image processing tasks, due to its high performance. Illustratively presented in Figure 6, its architecture encompasses a total of 13 convolutional layers and 3 fully connected layers. The convolutional layers, characterized by a kernel size of 3 by 3, are succeeded by a ReLU (rectified linear unit) activation function. Max-pooling is applied after every two convolutional layers to reduce the spatial dimensionality. The final layers are fully connected, and have a softmax activation function for classification.

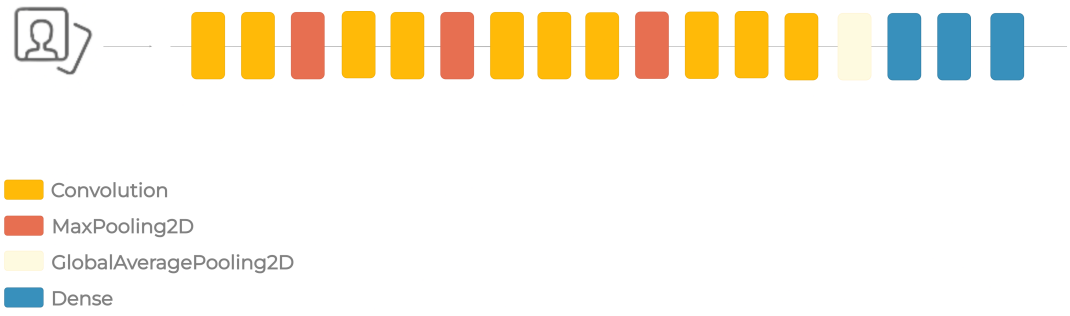


Figure 6: The architecture of VGG16 [36]

One of the reasons why the VGG16 architecture has become so popular is due to its simplicity and the use of small filter sizes. The use of small filters allows the network to learn local features, while the use of deeper layers enables the network to learn more complex features by combining the local features learned in earlier layers. This hierarchical structure of local and global features is what enables the VGG16 network to achieve high performance on various image recognition tasks. The choice of ReLU activation functions also improved the training time of the model by accelerating convergence.

However, it's worth noting that VGG16 does have its limitations. One drawback lies in its substantial number of trainable parameters, exceeding a staggering 138 million.Consequently, the model becomes more prone to overfitting, where it becomes overly specialized to the training data and performs poorly on unseen examples. Another limitation arises from the utilization of max pooling, which can inadvertently result in the loss of crucial spatial information. In the context of forgery detection, this loss is particularly undesirable, as it hampers the model's ability to accurately identify manipulated areas within an image. Furthermore, VGG16's reliance on small filter sizes brings forth an additional challenge. While these smaller filters allow for finer-grained analysis, they also possess a smaller receptive field. Consequently, they can only capture information from a limited number of neighboring pixels. The network may therefore struggle to capture higher-level features that span larger regions of the image. This is the reason why newer models, such as the previously stated InceptionResNetv2 and EfficientNetB4, have been proposed that use more effective receptive fields.

4.2.2 XceptionNet

Figure 7 shows the architecture of the XceptionNet model, a DCNN that has been used extensively for image feature extraction. The model was introduced as a variant of the Inception architecture, known for its efficient use of computational resources.

XceptionNet consists of 36 convolutional layers that are organized into 14 modules, all containing a sequence of depthwise separable convolutions and followed by a pointwise convolution. This design allows the network to learn efficient representations by decomposing standard convolutions into depthwise and pointwise convolutions to increase the power of the model, whilst minimizing the number of parameters.

The depthwise separable convolution consists of an input tensor that is, for each input channel, first convolved with a small filter. Followed by a pointwise convolution, a 1x1 convolutional layer is applied to mix the channels. This innovative technique supplants the conventional approach found in the Inception architecture, which employs a large filter that takes all input channels into account. The integration of depthwise separable convolutions into the XceptionNet architecture confers multiple advantages. Firstly, it reduces the number of parameters and computational demands imposed on the network. Consequently, training and inference processes accelerate, yielding efficiency gains. Secondly, the network attains heightened proficiency in capturing intricate spatial details by executing separate convolutions on each input channel. Lastly, the efficiency of the network improves as the computational cost. required for processing each input is reduced.

XceptionNet also harnesses skip connections, similar to the ResNet architecture, facilitate seamless

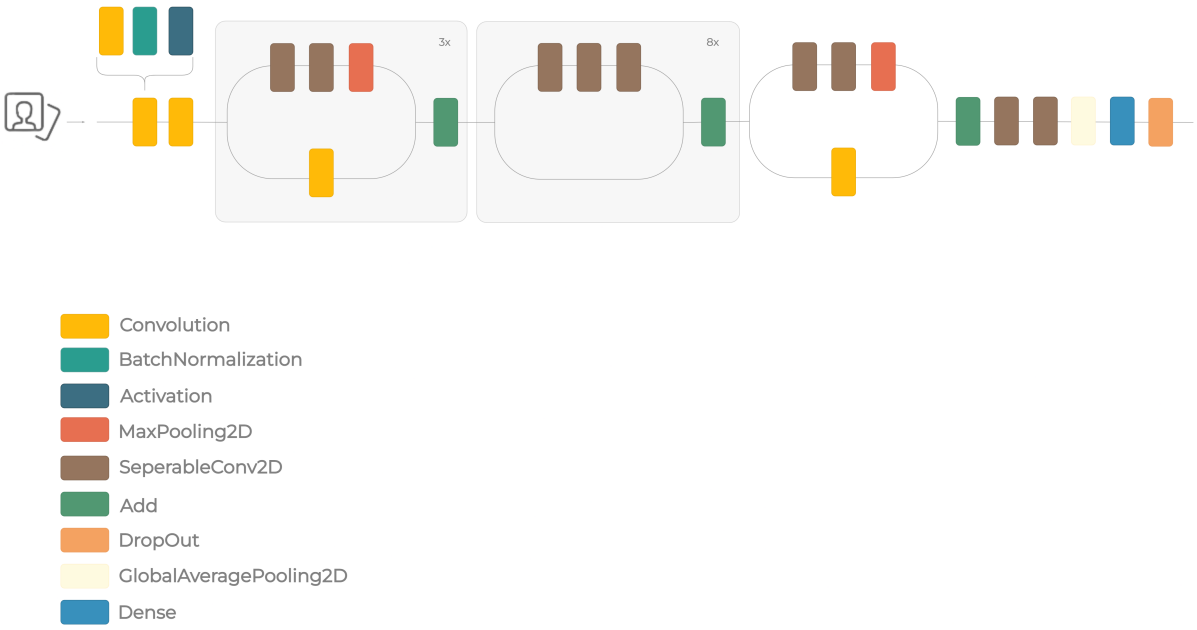


Figure 7: The architecture of XceptionNet [5]

information flow across the network while effectively mitigating the issue of vanishing gradients. Specifically, each module in XceptionNet encompasses a residual connection that bypasses the depthwise separable convolutions and merges the original input tensor with the module's output. This helps to ensure that the network can acquire nuanced and intricate features while avoiding excessive depth.

Another notable feature of XceptionNet is its use of batch normalization and ReLU activation functions after every convolutional layer, which can be seen in the first layer of Figure 7. Batch normalization normalizes the activations of each layer to have zero mean and unit variance, which helps to reduce the internal covariate shift and improve the stability and speed of training [21]. The ReLU activation function, on the other hand, introduces non-linearity into the network and helps the network capture complex patterns and features.

The incorporation of depthwise separable convolutions within the XceptionNet architecture offers a noteworthy advantage over the conventional convolutions employed in VGG16. While XceptionNet also uses small filters, this innovative technique empowers XceptionNet to learn finer details within the input data while significantly reducing computational demands. Notably, XceptionNet has remarkable results across multiple publicly available benchmark datasets, including the widely recognized ImageNet and COCO datasets. These achievements underscore the efficacy of XceptionNet in delivering high-quality results, however, the newer models EfficientNetB4 and InceptionResNetv2 have since surpassed XceptionNet in terms of accuracy and efficiency[3].

[3]Keras API, `https://keras.io/api/applications/`, accessed 02.2023

4.2.3 EfficientNetB4

EfficientNetB4 is a network that is part of the EfficientNet family. Which was designed to achieve superior accuracy while using fewer parameters than traditional CNNs. The results presented in Figure 8 demonstrate the networks of the EfficientNet family compared to other popular convolutional neural networks.

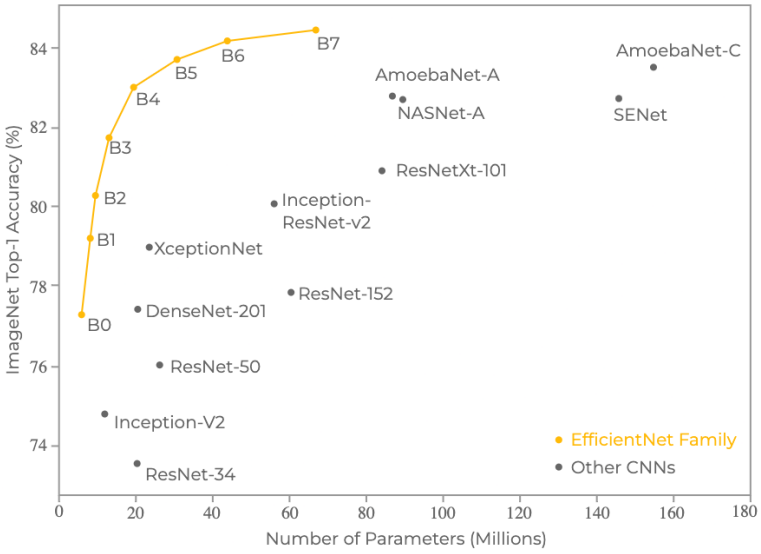


Figure 8: Comparison of the performance of EfficientNetB0-B7 on the ImageNet dataset and their number of parameters [39]

The EfficientNet family includes models that are optimized for trade-offs between performance and efficiency. EfficientNetB0 is the smallest and most computationally efficient, while EfficientNetB7 is the largest and most powerful. EfficientNetB4 is an intermediate model that strikes a balance between computational efficiency and performance. It provides better performance than the EfficientNetB0-B3 networks, while still being more computationally efficient than the larger models that are in the family. The architecture of EfficientNetB4 is imported via the `keras-efficientnets` module and is illustrated in Figure 9.

The EfficientNetB4 architecture consists mainly of mainly MBConv blocks and SE-blocks. The MBConv block, also used in the FDFtNet architecture shown in Figure 5, reduces the computational cost of the network while increasing its accuracy [35]. The SE-block is used to improve the network's expressive power by adapting the feature maps across channels [17].
EfficientNetB4 also utilizes a compound scaling method to achieve better accuracy with fewer parameters. This method scales up the networks depth, width, and resolution in a balanced manner to optimize its performance.

The EfficientNetB4 architecture has shown excellent performance on several benchmark datasets, including ImageNet and CIFAR-10 and it has outperformed other popular CNN architectures such as VGG16 and XceptionNet [39].

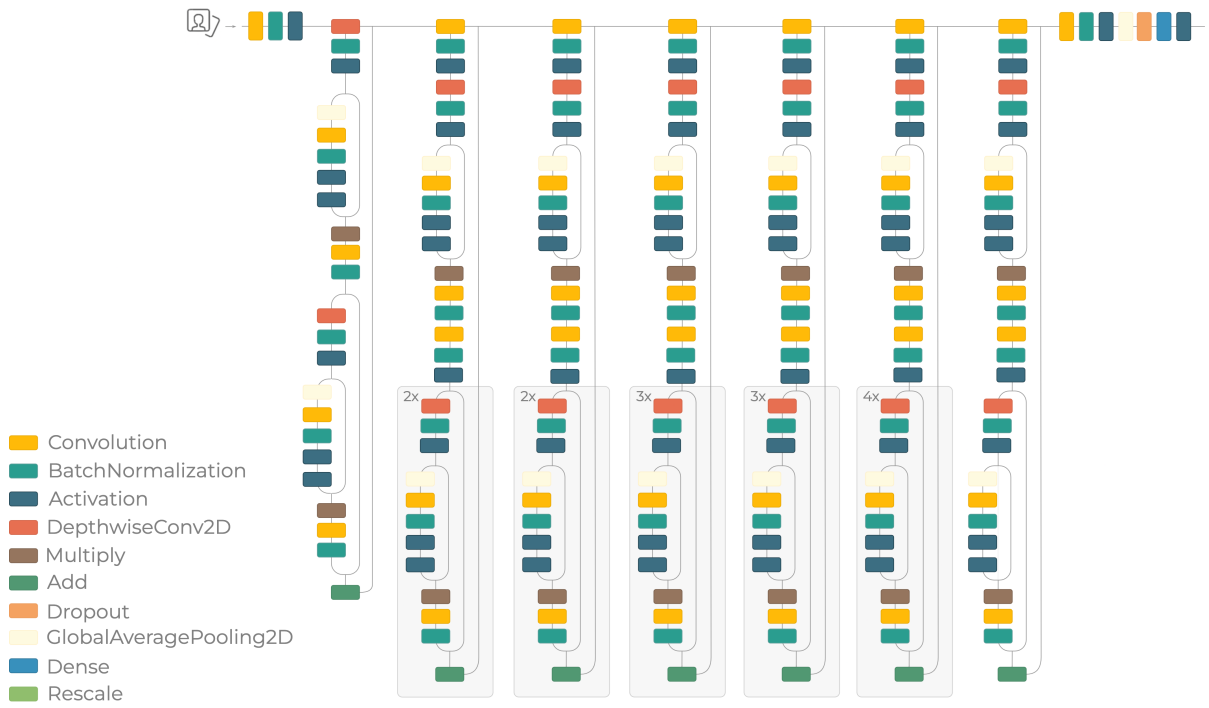


Figure 9: The architecture of EfficientNetB4 [39]

One reason for EfficientNetB4's superior performance is its efficient use of resources. EfficientNetB4 achieves similar or better accuracy in comparison to VGG16 and XceptionNet while using significantly fewer parameters. This efficient use of resources make EfficientNetB4 a better choice for feature extraction methods in scenarios where computational resources are limited, such as practical uses of image forgery detection. Another reason for EfficientNetB4's superior performance is its compound scaling method, which allows the network to adapt to a wide range of datasets.

4.2.4 InceptionResNet

The InceptionResNetv2 network (IRv2) is a hybrid deep learning model, that combines the strengths of the architecture of both InceptionNet and ResNet. The Inception module enables the network to capture features at multiple scales, while the residual connections from the ResNet module help the network to train deeper models without facing the vanishing gradient problem. Its success can therefore be attributed to the combination of multiple advanced techniques of two models into one model.

The architecture of IRv2 is illustrated in Figure 10 and is composed of repeated blocks of Inception-ResNet modules, followed by a global average pooling layer. Each Inception-ResNet module is composed of several parallel paths of convolutional layers with different kernel sizes, aimed at capturing features at different scales and concatenating their output features to form the final output of the module. These parallel paths can be seen in Figure 10, as well as the repeated blocks of 10 and 20 times. The residual connections from the ResNet module are incorporated by adding the output of the module to the input of the module, which is illustrated in the figure as well.

One of the strengths of IRv2 is its efficient use of parameters, allowing the model to be both efficient

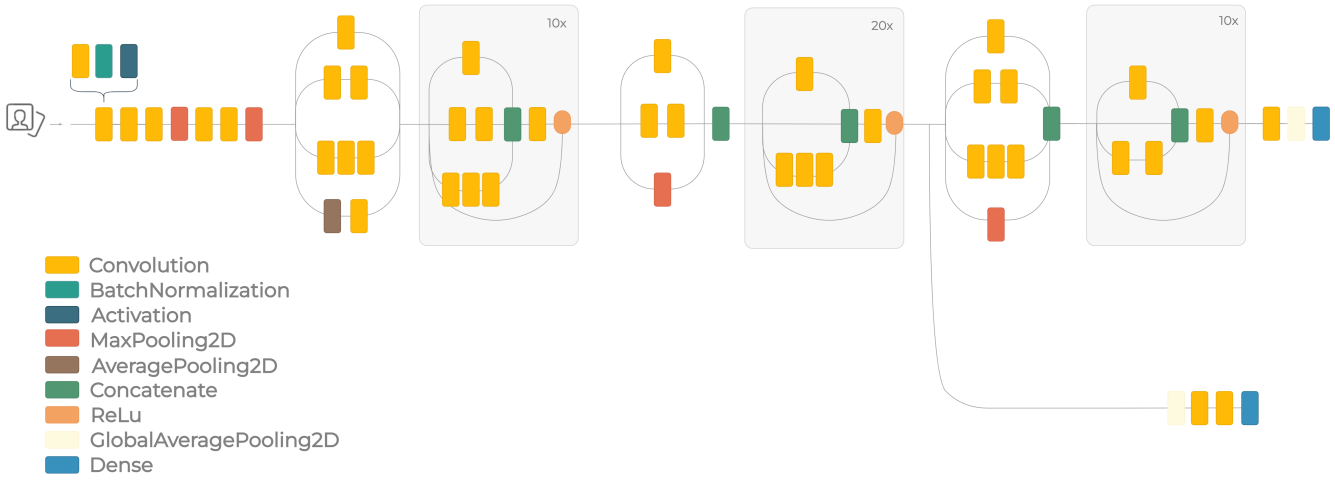


Figure 10: The architecture of InceptionResNet [37]

and effective. This is because the 1x1 convolutions in the Inception module reduce the number of input channels to each subsequent layer, thereby reducing the computational cost of the model. The IRv2 architecture also employs advanced regularization techniques such as batch normalization and dropout, to help prevent overfitting of the model. The use of a ReLU activation function also contribute to the effectiveness of the model.

One reason why the IRv2 architecture could produce better performance than both VGG16 and XceptionNet is due to its ability to capture features at multiple scales, a characteristic that is important for capturing forged features. Another reason for the success of the IRv2 architecture is its use of residual connections, that allow the model to train with deeper layers without facing the vanishing gradient problem.

4.3 Data

Each feature extraction model needs to be trained on data, and as previously stated, data containing ground truth labels per image. The feature extraction methods are pre-trained and fine-tuned on a novel dataset created by images of the FaceForensics++ dataset, CASIAv2.0, and the DEFACTO dataset. The datasets employed in this study were carefully selected based on the evaluation of a set of criteria, including their accessibility, the abundance of visual content they contain, and the diversity of tampering techniques represented within them.

The FaceForensics++ dataset is leading as a source for forged facial images, containing five different approaches of tampering with facial images, which was the reason for choosing this particular dataset. The different approaches included in this dataset are DeepFakes (from GitHub [4] and from the Deep Fake Detection Challenge (DFDC) [30], Face2Face [42], FaceSwap [23], NeuralTextures [41], and FaceShifter [24]. FaceForensics++ contains over 1000 forged videos of each manipulation method, containing individuals speaking in various languages and under different lighting conditions.

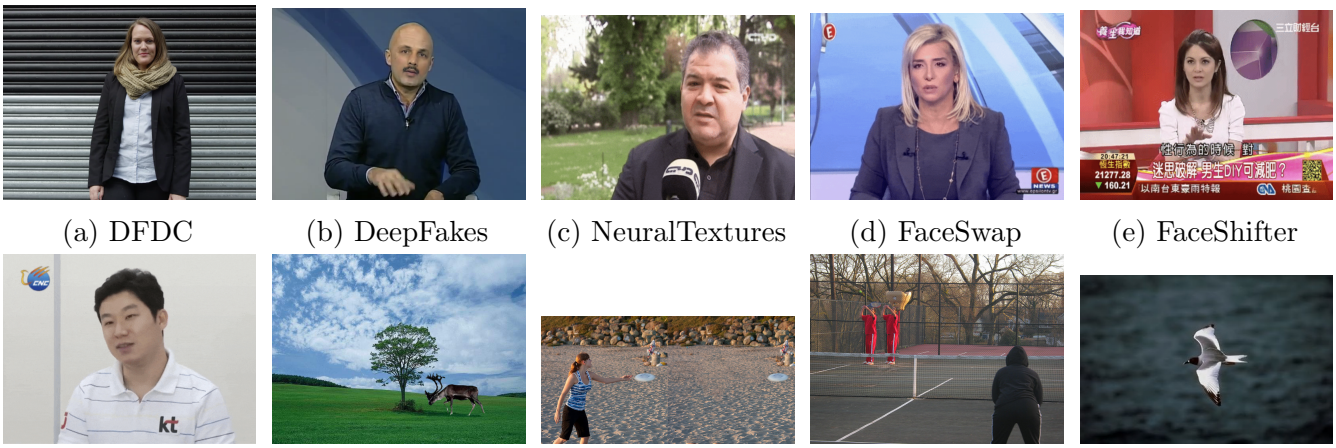
[4]DeepFakes GitHub https://github.com/deepfakes/faceswap

Furthermore, it contains 363 authentic (non-manipulated) videos that were obtained from YouTube.

The DEFACTO dataset was chosen for its extensive coverage of copy-move and inpainting manipulations in images. It contains over 200,000 images with various types of tampering, including copy-move, inpainting, and object removal. Copy-move forgery is the copying and moving an object of an image to a different part of the same image, while inpainting and removal concerns the filling in of missing parts or removal of an object in an image. The data was obtained from Kaggle [5] [6].

Similarly, the CASIAv2.0 dataset was chosen, which is a commonly used benchmark dataset for image forensics research. This dataset was chosen specifically for its representation of image splicing, a common form of tampering in which parts of one image are combined with another. The dataset contains over 10,000 images with splicing manipulations performed using different techniques on different image regions. The spliced images in this dataset were taken from a wide range of sources, including celebrity photos and natural scenery, and are diverse in terms of their content and quality, with variations in resolution, color, and lighting conditions. The CASIAv2.0 dataset was used to train the feature extraction model for splicing detection and was obtained from Kaggle as well [7].

It's worth noting that using different datasets for each type of manipulation helps ensure that the results of this study are not limited to a specific dataset. By fine-tuning the feature extraction models on multiple datasets with diverse tampering techniques, we can expect the models to be more robust and generalizable to different scenarios. This approach is particularly important in the field of forensic analysis, where tampering techniques are constantly evolving and improving.



(a) DFDC (b) DeepFakes (c) NeuralTextures (d) FaceSwap (e) FaceShifter

(f) Face2Face (g) Splicing (h) Inpainting (i) Copy Move (j) Real

Figure 11: Random example images of the dataset including all nine forgery types, where six are facial images and three are non-facial images

[5] DEFACTO Copy-Move dataset https://www.kaggle.com/datasets/defactodataset/defactocopymove
[6] DEFACTO Inpainting dataset https://www.kaggle.com/datasets/defactodataset/defactoinpainting
[7] CASIAv2.0 Dataset on Kaggle https://www.kaggle.com/datasets/sophatvathana/casia-dataset

5 Experiments

5.1 Prerequisites and hardware

The networks used in this project use different previous versions of various libraries, and the results are greatly depended on the exact version of each library. These libraries and their versions can be installed using their documentation. It is recommended to either use or create a virtual environment with Linux distributed operating system with Ubuntu 20.04 of *aarch64* architecture, as these networks appeared to be difficult to get working on MacOS or Windows. The used libraries and their corresponding versions can be found in Table 3.

Library/package	Version
Python	3.6.8
Tensorflow	1.13.1
Keras	2.2.4
Numpy	1.19.2
Pillow	8.3.1
Matplotlib	3.3.4
Keras Preprocessing	1.1.2
Keras Applications	1.0.8
Keras EfficientNets	0.1.7

Table 3. Required libraries and corresponding versions used in the training, testing, and visualization code of this study

The hardware used belongs to the Data Science Lab (DSLab) of the LIACS Research and Education Lab (REL). The machine used for the training and testing of the networks contains high-end GPUs and can specifically be used for CUDA-supported computations. The GPU of GeForce NVIDIA GTX TITAN X with 11.93GiB memory was used for the majority of the training procedure.

5.2 Data processing

As previously mentioned, sufficient datasets to train manipulation detection networks are rare. To improve the accessibility of the code and limit the amount of storage needed to run these networks, a script was created to load the videos of the FaceForensics++ dataset directly of their download script and cut out the frames of the video. A total of 500 videos per manipulation method was used and two frames per video were extracted. The copy-move and inpainting datasets from DEFACTO were downloaded using Kaggle, as well as the splicing data from CASIAv2. The images are rescaled to an 80x80 resolution to mimic the harshest JPEG compressions seen on the internet [22] and are displayed as a `Numpy ND array`. This array, along with its corresponding labels is shuffled by permutation and split into training, testing, validation, and fine-tuning sets by a ratio of 70:10:10:10. The number of images used from each dataset and the distribution of real and forged images is depicted in Table 4.

	Total	Train	Test	Validation	Fine-Tuning
Forged images					
		FaceForensics++			
FaceShifter	1000	700	100	100	100
NeuralTextures	1000	700	100	100	100
DeepFakes	1000	700	100	100	100
FaceSwap	1000	700	100	100	100
Face2Face	1000	700	100	100	100
		DEFACTO			
Copy-Move	4000	2800	400	400	400
Inpainting	4000	2800	400	400	400
		CASIAv2			
Splicing	2000	1400	200	200	200
Real images					
YouTube	726	507	73	73	73
MS COCO	2000	1400	200	200	200
Total	20726	14507	2073	2073	2073

Table 4. Number of images used from each dataset to form the novel dataset and their distribution over the training, testing, validation, and fine-tuning sets

5.3 Training

The majority of the training procedure has been replicated from the original research work [22]. Following this approach, the backbone models specified in Section 4.2 are initially pre-trained with the data illustrated in Section 5.2. Specifically, the pre-training phase encompasses 300 epochs, where each epoch comprises 55 steps. Every individual model undergoes training using the Adam optimizer, a stochastic gradient descent approach that relies on the adaptive estimation of first-order and second-order moments. The minimum number of steps per epoch is determined by the batch size of 128 and the total number of training images. During the training process, the network utilizes the `ImageDataGenerator` data generator, and Early Stopping is applied after the validation loss ceases to decrease for 40 epochs. The Early Stopping Patience was scaled up, as the network had a delay of improvement for the first epochs. To avoid overfitting, the drop-out rate is set to 0.2. Furthermore, the learning rate is initialized at 0.3 and is gradually adjusted with a factor of 0.1 via the implementation of the `ReduceLRonPlateau` algorithm by Keras.

As previously stated, a portion of the data is set aside for fine-tuning purposes, and this step is applied to each of the pre-trained networks. Fine-tuning is executed by subjecting the network to 300 epochs and the batch size, drop-out rate, and learning rate remain the same as those utilized during the pre-training phase.

5.4 Evaluation metrics

In this study, a comprehensive evaluation of various classification metrics was performed to determine the most appropriate metrics for evaluating the performance of the image forgery detection

model. After a thorough analysis, the metrics selected for this study included the F2-score, precision, recall and the area under the receiver operating characteristic curve (AUROC), as well as two novel metrics; the Detection Improvement Ratio (DIR) and Image Forgery Efficiency Ratio (IFER). In this section we will state the metrics that were used and explain them in detail, starting with the definition of the existing metrics F2-score, precision, recall, and AUROC.

In the field of binary classification models, the F_β-score in Equation 2, also known as the F1 score, is a widely used evaluation metric. It combines the precision and recall depicted in Equation 1.

$$\text{Precision} = \frac{TP}{TP + FP} \ , \qquad \text{Recall} = \frac{TP}{TP + FN} \tag{1}$$

Setting β to 2 in the F_β calculation places more emphasis on recall than precision. This choice is justified in image forgery detection studies where identifying as many actual forgeries as possible (high recall) is typically more critical than minimizing false positive detections (precision). By assigning a higher weight to recall, the F2 score with β set to 2 achieves a balance between these two metrics, enabling an evaluation that aligns with the specific needs of forgery detection. This emphasis on recall acknowledges the importance of minimizing the risk of missing any forged images, as even a single undetected forgery can have severe consequences. Hence, the F2-score proves to be a valuable and appropriate metric for assessing image forgery detection algorithms and was selected as one of the primary evaluation metrics.

$$\text{F}_\beta\text{-Score} = (1 + \beta^2) \times \frac{Precision \times Recall}{\beta \times (Precision + Recall)} \tag{2}$$

Additionally, the AUROC was included in the assessment of this study. The AUROC (Area Under the Receiver Operating Characteristic Curve) stands as a prominent evaluation metric highly regarded for its steadfastness and resilience. This metric quantifies the overall performance of a classification model by assessing the area under the curve derived from plotting the True Positive Rate (TPR), also known as recall, against the False Positive Rate (FPR). The Receiver Operating Characteristic (ROC) curve graphically represents this trade-off across various classification threshold values.
The AUROC is formulated in Equation 3, and the TPR and FPR can be calculated using the equations in Equation 4.

$$\text{AUROC} = \int_{-\infty}^{\infty} \text{TPR}(f(x)) \, \text{dFPR}(f(x)) \tag{3}$$

Due to its popularity amongst image forgery detection methods, the AUROC was included to account for a fair and clear comparison to other image forgery detection methods.

$$\text{TPR} = \frac{TP}{TP + FN} \ , \qquad \text{FPR} = \frac{FP}{FP + FN} \tag{4}$$

5.4.1 RDIR and DIR

The Detection Improvement Ratio (DIR), compares the accuracy of a given feature extraction method as a standalone approach to its accuracy when used as a backbone model in an image forgery detection algorithm. Specifically, DIR is defined as in Equation 5, where A_{IFD} denotes the accuracy of the feature extraction method when used as a backbone model in an image forgery detection algorithm defined by the F2-score, and A_{FEM} is the accuracy of the same feature extraction method as a standalone approach, defined by the top1-accuracy noted by Keras[8].

$$\text{DIR} = \frac{A_{IFD} - A_{FEM}}{A_{FEM}} \tag{5}$$

The DIR can be calculated for the new approaches XceptionNet, EfficientNetB4 and InceptionResNetv2, as well as for the baseline VGG16. To investigate whether the newer approaches improve the image forgery detection, we can calculate the Relative Detection Improvement Ratio (RDIR), which compares the DIR of a given feature extraction method x to the DIR of the baseline model.

$$\text{RDIR} = \text{DIR}_x - \text{DIR}_{base} \tag{6}$$

The rationale behind the RDIR metric is rooted in the premise that an effective feature extraction method should yield superior results when employed in an image forgery detection algorithm compared to a lower performing feature extraction method.

A high positive RDIR value signifies a substantial improvement of the new method over the old baseline method in both standalone accuracy and forgery detection accuracy, indicating its efficacy as a backbone model. An RDIR value of around zero demonstrates that the backbone model performs equally to the baseline model, where a small positive RDIR indicates a moderate improvement and a small negative RDIR indicates a slight decline in performance. Coversely, a high negative RIA value represents a significant deterioration in performance, where the new method performs significantly worse than the baseline method and may not be well-suited for the image forgery detection task when used in conjunction with the particular algorithm. The novel RDIR method and it therefore provides a direct measure of the improvement that is obtained when using a specific feature extraction method over an older method in image forgery detection.

5.4.2 IFER

The Image Forgery Efficiency Ratio (IFER) is a novel metric that was designed to assess the efficiency and performance trade-off of feature extraction methods within the context of image forgery detection. IFER offers a measure of the trade-off between the accuracy of detecting image forgeries and the computational cost required during the feature extraction process. The primary objective of IFER is to evaluate the effectiveness of feature extraction methods by considering their performance relative to the computational resources required.

[8]Keras API, https://keras.io/api/applications/, accessed 02.2023

IFER is defined as the harmonic mean of the accuracy (A) achieved by a feature extraction method in detecting image forgeries, the corresponding computational cost (C) associated with the training of the algorithm, and the time (T) required to classify an image. This is defined in Equation 7, where n is a variable for the calculation of the harmonic mean and is set to $n = 3$, representing the number of metrics in the equation. The variables A_r, C_r, and T_r denote the relative accuracy, computational cost, and time, respectively, and are defined in Equation 8. The weights were empirically determined and set to $w_1 = 0.6$, $w_2 = 0.2$, and $w_3 = 0.2$, due to the importance of the accuracy. In Equation 8, x denotes the value of the current network and b denotes the value of the best-performing network for that metric. The use of the harmonic mean in this calculation is scientifically justified due to its ability to emphasize smaller values and handle inverse relationships. By considering the reciprocals of the values, it effectively deals with the inverse relationships between accuracy, time, and computational cost. This metric captures the efficiency of the method in terms of accurately identifying image forgeries relative to the computational effort expended in extracting the features.

$$\mathrm{IFER}_x = \frac{n}{w_1 \frac{1}{A_r} + w_2 \frac{1}{C_r} + w_3 \frac{1}{T_r}} \tag{7}$$

$$\mathrm{A}_r = \frac{A_x}{A_b} \ , \qquad \mathrm{C}_r = \frac{C_b}{C_x} \ , \qquad \mathrm{T}_r = \frac{T_b}{T_x} \tag{8}$$

A higher IFER value indicates a more efficient feature extraction method in image forgery detection, as it achieves a greater accuracy in identifying forgeries while minimizing the computational resources consumed. This metric aids in the identification of feature extraction methods that strike a favorable balance between accuracy, computational cost, and the temporal aspect, making them suitable choices for real-time image forgery detection systems or scenarios with limited computational capabilities.

6 Results

The study presents three categories of results, which will be shown in this section. Firstly, three training metrics of the pre-training and fine-tuning process of the four backbone models were evaluated. These training metrics encompass the accuracy, loss, and learning rate during training. These metrics provide an understanding of the training process and helped to identify problems that require attention. Secondly, the performance of the fine-tuned model was assessed on the test data, which provides results on the model's ability to generalize beyond the training data and is essential for any machine learning study, as well as the results for the metric RDIR. Finally, the classification times of the four backbone networks were analyzed during the training, as well as the classification of testing single images, and the results of the metric IFER, which provides valuable information on the computational efficiency of the models.

Figures 12, 13, and 14 provide a graphical representation of the metrics observed during the pretraining and fine-tuning process, and illustrate the accuracy, loss, and learning rate, respectively. In these figures, the accuracy was measured by the proportion of correctly classified images. The loss metric is composed of the error rate stated by Keras. The learning rate is a critical parameter that is set in the training code and it determines the step size of the optimization algorithm.

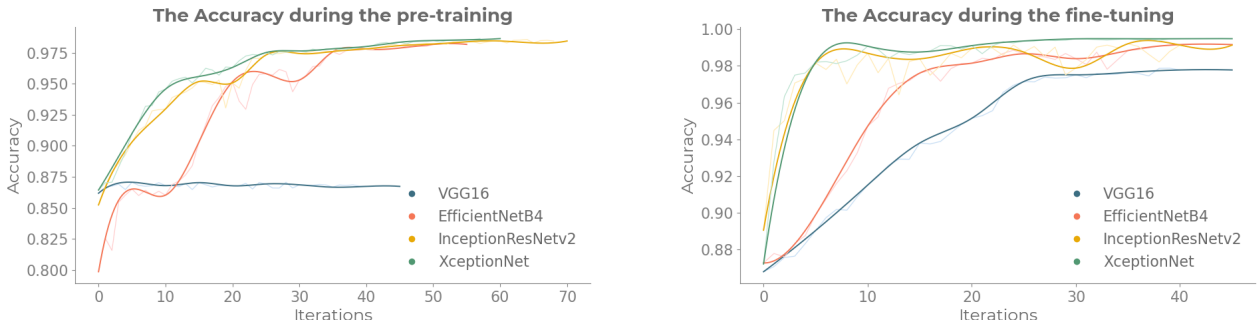


Figure 12: The accuracy of the feature extraction methods VGG16, EfficientNetB4, InceptionResNetv2, and XceptionNet during the pre-training and finetuning of the methods as backbone networks for image forgery detection method FDFtNet

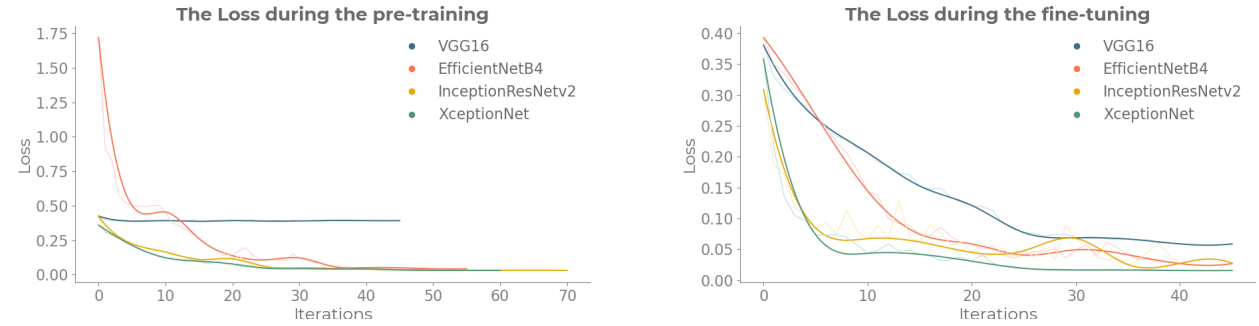


Figure 13: The loss of the feature extraction methods VGG16, EfficientNetB4, InceptionResNetv2, and XceptionNet during the pre-training and finetuning of the methods as backbone networks for image forgery detection method FDFtNet

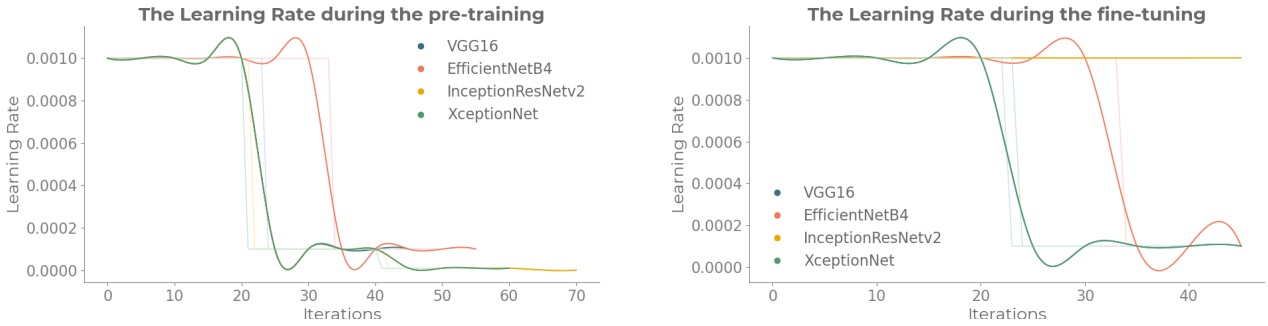


Figure 14: The evolution of the learning rate of the feature extraction methods VGG16, EfficientNetB4, InceptionResNetv2, and XceptionNet during the pre-training and finetuning of the methods as backbone networks for image forgery detection method FDFtNet

Section 5.4 discussed the evaluation metrics used for the test data, which encompassed the F2-score, AUROC, precision, and recall. These metrics, as well as the results of the novel metrics RDIR and IFER are presented in Table 5. The precision, recall, and F2-Score are the macro average of these metrics on the binary labels. The DIR and IFER scores of the feature extraction methods were also illustrated using a graph, which can be seen in Figure 15. The results were also categorized in

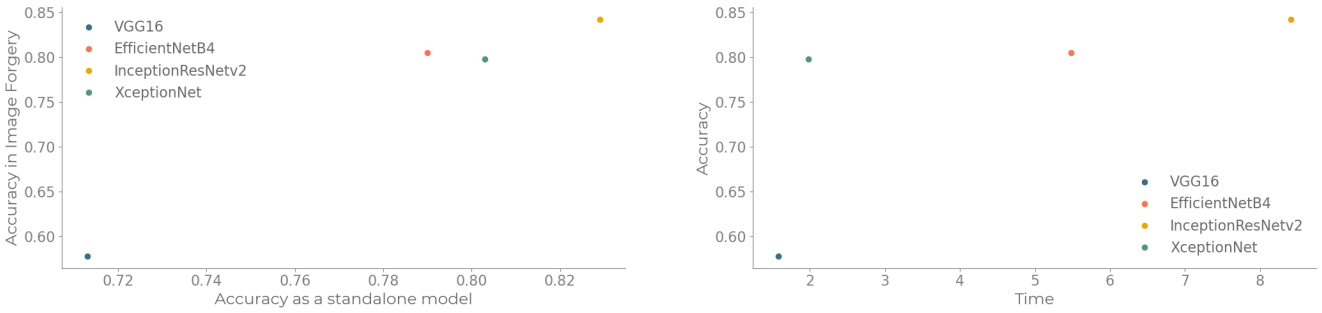
the different types of forgery as was shown in Table 4, which can be seen in Table 6.

Network	AUROC	Precision	Recall	F2-Score	RDIR	IFER
VGG16	0.500000	0.43	0.50	0.57796	0.0	2.09799
XceptionNet	0.766744	0.67	0.62	0.80504	0.20843	2.78144
InceptionResNetv2	0.768747	0.70	0.65	0.84259	0.23869	1.96183
EfficientNetB4	0.766489	0.67	0.61	0.79824	0.15229	1.39595

Table 5. The results of the four feature extraction methods on the novel dataset for evaluation metrics AUROC, Precision, Recall, F2-score, and novel metrics RDIR, comparing all methods to VGG16 and IFER, the tradeoff for efficiency and performance

	VGG16	XceptionNet	InceptionResNetv2	EfficientNetB4
Forged images				
FaceShifter	1.00	1.00	1.00	1.00
NeuralTextures	1.00	1.00	1.00	1.00
DeepFakes	1.00	1.00	1.00	1.00
FaceSwap	1.00	1.00	1.00	1.00
Face2Face	1.00	1.00	1.00	1.00
Copy-Move	1.00	0.49	0.56	0.53
Inpainting	1.00	0.50	0.53	0.49
Splicing	0.32	0.57	0.58	0.58
Real images	0.25	0.61	0.66	0.65

Table 6. The macro average of the f2-score of the four feature extraction methods categorized by the forgery types present in the novel dataset, where the importance of the accuracy on real images should be noted



(a) DIR

(b) IFER

Figure 15: The DIR and IFER of the four feature extraction models, where the DIR is the correlation between the accuracy of the standalone model and the accuracy in image forgery detection, and the IFER denotes the correlation between time and accuracy

To assess the efficiency of each backbone network, an analysis of the training and classification times for each model was performed. The results highlight the efficiency of each backbone network in training and classifying new images and can provide valuable insights for researchers and practitioners in the field of image processing. The results of this analysis are presented in Table 7.

Network	Pre-training	Fine-tuning	Classification	Computational Cost in FLOPs
VGG16	24.93 min	40.85 min	1.578 ms	1.26 billion
XceptionNet	41.01 min	47.86 min	1.983 ms	0.71 billion
InceptionResNetv2	59.77 min	51.85 min	8.409 ms	1.71 billion
EfficientNetB4	59.21 min	67.46 min	5.481 ms	1.52 billion

Table 7. The temporal aspect and computational costs of the feature extraction methods VGG16, XceptionNet, InceptionResNetv2, and EfficientNetB4, where the training and classification times are denoted in minutes and milliseconds, respectively, and the computational cost is denoted by the total number of FLOPs in billions.

7 Discussion

This study evaluated the performance of four different feature extraction methods - VGG16, XceptionNet, InceptionResNetv2, and EfficientNetB4 - in detecting image forgeries. The hypothesis proposed that newer feature extraction methods would outperform older methods. The network was trained on a dataset of over 14.000 images and tested on a separated dataset of 2.072 images. The results showed that newer feature extraction methods did indeed perform better than older methods, lending support to the hypothesis. Additionally, the use of the novel dataset in this study enhances its generalizability and robustness. The diversity that is captured in the dataset is crucial for training and testing image forgery detection algorithms that need to generalize across a wide range of conditions. Moreover, the larger sample size provided by the dataset contributes to more robust statistical analysis and increased dataset representativeness. The introduction of new combinations of objects or scenes that were not present in the individual datasets further tests the generalizability of image forgery detection algorithms.

Specifically, the results showed that VGG16, considered the baseline model as it is an older feature extraction method, performed relatively poorly, with an f2-score of 0.58 and a complete inability to detect real images. Further analysis of Table 6, indicates that VGG16 exhibits superior performance compared to other networks in terms of detecting forged images. However, VGG16 tends to misclassify a substantial portion of the authentic images as forged, resulting in an AUROC score of 0.5000. Consequently, this finds VGG16 as the algorithm with the poorest performance among the evaluated approaches. The suboptimal performance of VGG16 can be attributed to the limitations of VGG16. In particular, the smaller filter sizes that cause the inability to capture fine details within the sample images. VGG16 is a relatively shallow architecture and it relies heavily on pooling layers to reduce the dimensionality of the feature space. VGG16 was a popular and groundbreaking architecture when it was introduces, but it was primarily designed for the task of image classification. Luckily, there exist newer feature extraction methods that have

surpassed the performance of VGG16, which are a better choice for the use in image forgery detection.

In contrast, the newer feature extraction methods - XceptionNet, InceptionResNetv2, and EfficientNetB4 - all performed significantly better. EfficientNetB4 achieved an overall f2-score of 0.79824 and a score of 0.65 for detecting real images. XceptionNet performed slightly better, achieving an overall f2-score of 0.80504 and an f1-score of 0.61 for detecting real images, whereas InceptionResNetv2 achieved the highest performance, with an f2-score of 0.84259, and a high ability to detect real images, with a score of 0.66.

The superior performance of the newer feature extraction methods can be attributed to two potential reasons. Firstly, these methods have deeper and more complex architectures compared to VGG16. The increased depth allows the networks to capture more intricate features in images, which is particularly important for detecting image forgeries, as subtle differences can be indicative of tampering. Secondly, the newer feature extraction methods incorporate novel architectural features that improve the quality of the features extracted. To name a few, XceptionNet uses depthwise separable convolutions, which allows for more efficient and effective feature extraction. InceptionResNetv2 is an architecture that combines the strengths of the inception module with residual connections, that creates the unique ability to improve gradient flow and efficiently use resource utilization. EfficientNetB4 uses a novel scaling method that optimizes the balance between model depth, width, and resolution.

However, it is worth noting that although the newer feature extraction methods performed better than the older method VGG16, they all struggled to detect real images. This result may be due to the fact that real images are increasingly difficult to detect, as they are by definition not manipulated in any way. In addition to the overall scores of the feature extraction methods, another notable characteristic is that the feature extraction methods seem to be performing better on the facial images than that of copy-move, inpainting, and splicing. This may be because the dataset contains imbalanced classes, with more forged facial images than real ones.

The observed results reveal a notable disparity between the AUROC of XceptionNet, which stood at 76.67%, and the previously reported a remarkable accuracy of 99.37% in the original paper on FDFtNet [22]. The factors contributing to this discrepancy remain unclear. It is plausible that the variances in results could be attributed to variations in the training dataset size or the utilization of a specific dataset instead of a generalized and robust dataset as was done in this study. Reproducing the results from the original FDFtNet paper, which would yield a significant increase of 22.7%, suggests that advanced feature extraction methods such as EfficientNetB4 and InceptionResNetv2 would potentially achieve an accuracy of at least 99.57%. This is in agreement with the novel RDIR metric, shown in Table 5, which revealed that XceptionNet, InceptionResNetv2, and EfficientNetB4 received an RDIR score of higher than zero when compared against VGG16. This indicated that these methods perform better than the older feature extraction method VGG16, and the RDIR shows that this could have been predicted using their accuracies as standalone models. As expected, VGG16 received an RDIR of 0.0 compared to itself. Furthermore, state-of-the-art methods like ManTraNet and BusterNet, currently employ VGG16 as the feature extraction model. These methods could benefit significantly from incorporating the newer feature extraction techniques, potentially leading to an additional accuracy boost of 34.49%. This substantial enhancement could

pave the way towards achieving near-perfect accuracy levels, approaching 100%.

Moreover, the examination of Table 7 and Figure 15 provides a comprehensive overview of these classification times, showcasing VGG16 as the fastest model, closely followed by XceptionNet. EfficientNetB4 ranks third in terms of speed, while InceptionResNetV2 exhibits the longest classification time of 8.409ms. The results of the IFER metric reveal a novel correlation between the accuracy of the models and their respective computational and temporal costs. The XceptionNet model seems to perform best regarding accuracy in combination with the time and computational cost with an outstanding IFER of 2.78144. The VGG16 model, with its shallow network, receives an IFER of 2.09799, closely followed by InceptionResNet with an IFER of 1.96183. The similarity in values can be attributed to the trade-off between performance and computational efficiency. The VGG16 model is fast but performs poorly, while the InceptionResNet model is slower but exhibits superior performance. The EfficientNet model performs the poorest due to its IFER of 1.39595, which can be explained by its high computational cost and classification times.

Considering the practical application of these models in social media or news platforms, the classification times hold significant relevance. For instance, the social media platform Instagram sees a staggering 95 million pictures posted per day [9]. The computational energy required to classify all of these images before publication can be immense. A relevant example from Instagram in 2022 involved the placement of a label redirecting all pandemic-related images to a source with credible information. Similarly, labeling potentially forged images with informative content, such as "This image is possibly forged, click here for more information on image forgery and guidelines for assessing credibility," could substantially aid in preventing the spread of misinformation.

8 Conclusions and Further Research

Our study focused on feature extraction, which is a critical part in the detection of image forgery. We used four different feature extraction methods as backbone architectures and trained and fine-tuned them on our novel dataset of real and manipulated images. Our aim was to identify whether newer feature extraction methods can improve image forgery detection.

To achieve our objective, we first evaluated the pretraining and fine-tuning stages whilst training the four backbone models. The pretraining involved using a large dataset to initialize the model's parameters, while the fine-tuning was done on a smaller dataset that was seperated (the fine-tuning dataset) to improve the model's performance. The second phase of our study encompassed the evaluation of the fine-tuned model on our test dataset. The test set was used to evaluate the performance of our fine-tuned model.

When analysing the results, it can be concluded that the newer feature extraction methods selected for this study, XceptionNet, EfficientNetB4, and InceptionResNetv2, outperform the older method VGG16 in image forgery detection. This conclusion aligns with the hypothesis that newer feature extraction methods achieve better performance when used in existing image forgery detection methods.

This thesis presents four significant contributions to the field of image forgery detection. Firstly, it conducts an extensive study that specifically focuses on the feature extraction stage of image forgery detection, rather than solely emphasizing overall accuracy. By highlighting the importance of feature extraction methods, this thesis provides valuable insights for researchers to make informed choices in selecting feature extraction techniques. The findings of this study demonstrate that superior performing feature extraction methods lead to improved performance in image forgery detection. Secondly, this study introduces a novel metric called the Detection Improvement Ratio (DIR). The DIR metric establishes a correlation between the overall performance of a feature extraction method and its influence on the accuracy of an image forgery detection algorithm. This metric serves as a valuable tool for evaluating and comparing different feature extraction methods in the context of image forgery detection. Furthermore, this thesis employs a novel dataset to ensure that the obtained results are not limited to a specific dataset but rather generalize well across various scenarios. This novel dataset significantly contributes to the field by increasing the diversity of the data and generating new combinations of objects or scenes that were not present in individual datasets. Consequently, the dataset enhances the reliability and robustness of the study's findings. Lastly, this research introduces a novel metric called the Image Forgery Efficiency Ratio (IFER). The IFER metric combines the accuracy of image forgery detection algorithms with the time required to classify a testing image. This metric addresses the practical usage of image forgery detection algorithms by considering their efficiency and real-world applicability. In summary, Through a thorough investigation of feature extraction, the introduction of the Detection Improvement Ratio and the Image Forgery Efficiency Ratio metrics, and the use of a unique dataset, this study makes significant contributions to the field of image forgery detection. These contributions advance our understanding of image forgery detection methods and provide valuable tools for researchers and practitioners in this domain.

It is important to note that the performance of the four previously stated feature extraction methods depend on the size of the dataset, as well as the representation of the images in the dataset. This thesis has a few limitations, as many networks and datasets are not publicly available. For example, some training data of state-of-the-art networks that use feature extraction methods mentioned in this thesis were not available for further research.

In conclusion, this study provides interesting insights into the performance of different feature extraction methods in image forgery detection and our findings can serve as a cornerstone for future research into improving existing image forgery detection models by using different feature extraction methods. The results suggested that newer feature extraction methods provide higher performance compared to older methods. These feature extraction methods can be used to improve the accuracy of existing image forgery detection systems, and the contributions made in this study advance our understanding of image forgery detection methods. Future research can build on these findings by investigating the performance of a wider range of different feature extraction methods on image forgery detection tasks, as well as exploring different networks other than FDFtNet and using a larger training and testing dataset, as the feature extraction methods had trouble correctly classifying real images.

References

[1] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. In *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7, 2018.

[2] Y. Akbari, N. Almaadeed, S. Al-Maadeed, F. Khelifi, and A. Bouridane. Prnu-net: a deep learning approach for source camera model identification based on videos taken with smartphone. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 599–605, 2022.

[3] S. S. Ali, I. I. Ganapathi, N.-S. Vu, S. D. Ali, N. Saxena, and N. Werghi. Image forgery detection using deep learning by recompressing images. *Electronics*, 11(3):403, 2022.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Lecture notes in computer science*, 3951:404–417, 2006.

[5] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *Computing Research Repository (CoRR)*, 2017.

[6] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[7] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li. MVSS-net: Multi-view multi-scale supervised networks for image manipulation detection. In *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14, 2022.

[8] J. Dong, W. Wang, and T. Tan. Casia image tampering detection evaluation database. In *Proceedings of the IEEE China Summit and International Conference on Signal and Information Processing*, pages 422–426, 2013.

[9] J. Flynn. Instagram Statistics [2023]: Facts About This Important Marketing Platform. https://www.zippia.com/advice/instagram-statistics/, 2023.

[10] T. Gloe and R. Böhme. The Dresden Image Database for benchmarking digital image forensics. In *Proceedings of the ACM Symposium on Applied Computing*, pages 1584–1590, 2010.

[11] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 63–72, 2019.

[12] Z. Guo, G. Yang, J. Chen, and X. Sun. Fake face detection via adaptive manipulation traces extraction network. *Computer Vision and Image Understanding*, 204:103170, 2021.

[13] L. Haodong, L. Weiqi, Q. Xiaoqing, and H. Jiwu. Image forgery localization via integrating tampering possibility maps. *IEEE Transactions on Information Forensics and Security*, 12(5):1240–1252, 2017.

[14] 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*, pages 770–778, 2016.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 630–645. Springer, 2016.

[16] A. Howard, M. McDonald, M. Stamm, and P. Bestagini. IEEE's Signal Processing Society - Camera Model Identification, 2017.

[17] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[18] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia. Span: Spatial pyramid attention network for image manipulation localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 312–328. Springer, 2020.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[20] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Computing Research Repository (CoRR)*, 2015.

[22] H. Jeon, Y. Bang, and S. S. Woo. Fdftnet: Facing off fake images using fake detection fine-tuning network. In *Proceedings of the International Conference in ICT Systems Security and Privacy Protection*, pages 416–430. International Federation for Information Processing (IFIP), Springer, 2020.

[23] M. Kowalski. FaceSwap. https://github.com/MarekKowalski/FaceSwap/, 2016.

[24] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5074–5083, 2020.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[26] X. Liu, Y. Liu, J. Chen, and X. Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *Proceedings of the IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7505–7517, 2022.

[27] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[28] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. Ieee, 1999.

[29] G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and M. Pic. Defacto: Image and face manipulation dataset. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 1–5, 2019.

[30] Y. Mirsky and W. Lee. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41, 2021.

[31] T.-T. Ng, J. Hsu, and S.-F. Chang. Columbia image splicing detection evaluation dataset. 2009.

[32] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4780–4789, 2019.

[33] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2019.

[34] R. Salloum, Y. Ren, and C.-C. J. Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[37] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet, and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *Computing Research Repository (CoRR)*, abs/1409.4842, 2014.

[39] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 6105–6114. PMLR, 2019.

[40] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. S. Woo. Gan is a friend or foe? a framework to detect various fake face images. In *Proceedings of the ACM/SIGAPP Symposium on Applied Computing*, pages 1296–1303, 2019.

[41] J. Thies, M. Zollhöfer, and M. Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[42] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.

[43] S. Walia and K. Kumar. Digital image forgery detection: a systematic scrutiny. *Australian Journal of Forensic Sciences*, 51(5):488–526, 2019.

[44] J. Wang, Z. Wu, J. Chen, X. Han, A. Shrivastava, S.-N. Lim, and Y.-G. Jiang. Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2364–2373, 2022.

[45] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler. Coverage — a novel database for copy-move forgery detection. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 161–165, 2016.

[46] Y. Wu, W. Abd-Almageed, and P. Natarajan. Busternet: Detecting copy-move image forgery with source/target localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 168–184, 2018.

[47] Y. Wu, W. AbdAlmageed, and P. Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2019.

[48] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[49] P. Zhou, B.-C. Chen, X. Han, M. Najibi, A. Shrivastava, S.-N. Lim, and L. Davis. Generate, segment, and refine: Towards generic manipulation segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13058–13065, 2020.

[50] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1053–1061, 2018.

[51] 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*, pages 8697–8710, 2018.