Vietnam Journal of Science and Technology ***62*** *(3) (2024) 562-576*

doi:10.15625/2525-2518/18626

Preserving authenticity: transfer learning methods for detecting and verifying facial image manipulation

Kinjal R Sheth[1, *], Vishal S Vora[2, *]

[1]*Assistant Professor, Electronics & Communication, L D College of Engineering, Ahmedabad - 380052 Gujarat, India*

[2]*Head & Associate Professor, Atmiya Institute of Technology, Atmiya University, Gujarat, India*

[*]Emails: *1.15115819007@atmiyauni.edu.in;krsheth@ldce.ac.in, 2. vishal.s.vora@gmail.com*

Received: 7 August 2023; Accepted for publication: 16 December 2023

Abstract. Facial retouching in supporting documents can have adverse effects, undermining the credibility and authenticity of the information presented. This paper presents a comprehensive investigation into the classification of retouched face images using a fine-tuned pre-trained VGG16 model. We explore the impact of different train-test split strategies on the performance of the model and also evaluate the effectiveness of two distinct optimizers. The proposed fine-tuned VGG16 model with "ImageNet" weight achieves a training accuracy of 99.34 % and a validation accuracy of 97.91 % over 30 epochs on the ND-IIITD retouched faces dataset. The VGG16_Adam model gives a maximum classification accuracy of 96.34 % for retouched faces and an overall accuracy of 98.08 %. The experimental results show that the 50 % - 25 % train-test split ratio outperforms other split ratios mentioned in the paper. The demonstrated work shows that using a Transfer Learning approach reduces computational complexity and training time, with a max. training duration of 39.34 min for the proposed model.

Keywords: Adam, Fine-tuning, VGG16, RMSprop, Retouching.

Classification numbers: 4.7.4, 4.8.2, 4.8.4, 4.10.2

1. INTRODUCTION

In our modern life, digital images have become indispensable. Unfortunately, the widespread availability of advanced image processing tools on the Internet has led to a proliferation of fake images. While some of these images may seem harmless, they have been exploited for nefarious purposes, such as creating counterfeit legal documents, manipulating evidence in legal proceedings, and distorting historical events. Furthermore, the prevalence of retouched images on social media platforms, often using filters to create flawless appearances, has fostered unrealistic beauty standards. Beauty and celebrity magazines also contribute to this phenomenon, perpetuating unrealistic expectations by showcasing heavily altered appearances, as depicted in Figure 1.

Image forgery poses a significant challenge, as it can be visually imperceptible when executed with precision. Reference [1] demonstrated how such alterations can negatively impact individuals' self-esteem by promoting unrealistic beauty standards. The introduction of the Photoshop Law in Israel further emphasizes the need for algorithms to detect tampering, reflecting the prevalence of this issue [2]. Moreover, beyond health and moral concerns, synthetic alterations affect biometric systems, potentially hindering accurate identification and auto-matching of bonafide faces[3].

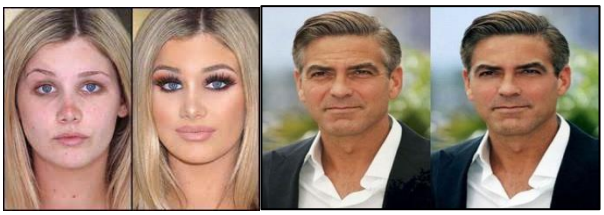


Figure 1. Showcasing the examples of facial tempering using Photoshop. The first image is real and the second image of the same person is retouched.

In order to maintain the integrity and authenticity of visual content, this research attempts to create an effective framework for the detection and classification of facial retouching using a transfer learning approach. This requires building a strong computational framework to recognize minute modifications made to facial images.

1.2 Literature review

Facial image retouching, photo spoofing or morphing, and makeup detection are widely studied areas and considered equivalent to detecting retouching. An SVR (support vector regression) between the altered and real photos was discovered in earlier studies by Reference [4]. In 2015, to assess whether makeup is present, Reference [5] suggested SVM and alligator classifiers to detect makeup using shape and texture features that are retrieved from the complete face. In 2016, Reference [6] used a supervised Boltzmann algorithm to detect retouching on an ND-IIITD retouched faces dataset. The dataset introduced contains 2600 real face images and 2275 face images which are retouched by the Portrait Pro max photo editing tool. The geometric and photometric features are used to train SVR for classifying retouched images. Hence, a total of 4 facial patches are used to detect retouching.

In 2017, Reference [7] introduced a new dataset, namely MDRF (multi demographic retouched dataset) containing the real and retouched face images of three ethnicities - Caucasian, Chinese, and Indian. The classification of retouched images is done using semi-supervised autoencoders. The model is trained on 4 patches of face images. An algorithm that recognizes altered face photos is presented by [8] and uses a gradient-based classification approach. In 2018, CNN (convolutional neural network) architecture was introduced to detect retouching on the standard ND-IIITD dataset [8]. Different non-overlapping face patches of size (128×128) and (64×64) are used to detect retouching. The classifiers used are SVM and Thresholding. In 2019, Reference [9] used 5 different photo editing tools to retouch 800 bonafide face images and detected retouching using a PRNU (photo-response non-uniformity) scheme. The PRNU-based detection scheme demonstrates robust discrimination between unaltered bonafide images and retouched images, achieving an average detection equal error rate of 13.7 %. In order to detect morphing faces, a physical reflection model is introduced that calculates the direction of light sources for the nose and eye regions [10]. Biological signals

(photo plethysmography) and 129 picture quality features were employed to detect fake images as binary classification [11, 12].

In 2023, an improvised patch-based deep convolutional neural network (IPDCN2) was presented in Reference [13], which effectively classifies facial images as either original or retouched through three stages: pre-processing using facial landmarks, high-level feature extraction with a CNN based on residual learning, and classification using fully-connected layers. The experimental results achieved an accuracy of 99.84 % (patch-based) on the ND-IIITD dataset and classification accuracies of 95.80 %, 83.70 %, and 97.30 % on the YMU, VMU, and MIW makeup datasets, respectively. Deep learning is a significant AI accomplishment [14]. Convolutional neural networks (CNN) are a common type of deep learning architecture [15]. Transfer learning helps avoid reinventing the wheel and lowers the cost of learning. Several extensively used pre-trained models include VGG16, VGG19, ResNet50, Inceptionv3, and EfficientNet [16].

Reference [17] introduces a retouching-FFHQ dataset, specifically used for detecting retouching. The TP, TN and accuracy of binary classification are analyzed using multi-granularity attention modules and compared over different transfer learning models such as VGG16, InceptionV3, ResNet50, DenseNet121, and EfficientNet. In Reference [18], transfer learning was employed for a classification task using pre-trained models (MobileNet V2, ResNet50, and VGG19), with VGG19 achieving the highest classification accuracy (95 %) and f1-score on a previously unseen dataset, despite a longer execution time of 7 hours, 5 minutes, and 52 seconds. The pre-trained VGG16 architecture was utilized for skin cancer image classification [19], exploring different color scales (HSV, YCbCr, and Grayscale). The evaluation shows that a classification accuracy of 84.242 % was achieved by a dataset created from RGB and YCbCr images. The research extracted feature parameters from different layers and analyzed VGG16's performance across color scales to determine how effective it is in classifying diseases.

Moreover, very little research has been carried out till now over the face images which are retouched using photo editing tools. When employing a DL (deep learning) model to identify retouching on facial photos[20], there are many difficulties as presented in Reference [21]: To train the model to recognize retouching accurately, a large number of images, well-labeled metadata, and a facial dataset comprising both legitimate and manipulated photo images are required. In this context, Transfer learning (TL) addresses various challenges and enables optimal detection accuracy [22]. TL offers several advantages in machine learning and deep learning tasks, including reduced training time, lower data requirements, and improved generalization.

Our contribution is as follows:

- For the proposed work, VGG16 with ImageNet weight is used as it gives a top-5 error of 9.3 % [23].
- For detecting retouching, an ND-IIITD retouched faces dataset is used. The dataset is divided into 80 % - 20 %, 70 % - 30 %, 60 % - 40 %, and 50 % - 50 % train-test split ratios.
- Two distinct 1st order optimizers, *Adam* and *RMSprop*, are used during fine-tuning.
- A total of 8 distinct experiments are performed over the proposed TL model and the classification accuracy of the model is evaluated for the different train-test split ratios.
- To the best of the authors' knowledge, no prior research has conducted timing analysis for training a model, an aspect that is evaluated in this study.

The flow of this research is stacked as follows: The proposed methodology, brief of VGG16 architecture, optimizers and facial dataset are outlined in Section 2. Result analysis of the proposed models is summarized in Section 3. Conclusions and future work are discussed in Section 4.

2. PROPOSED METHODOLOGY

In this work, we proposed a TL method to classify bonafide (real) vs fake (retouched) face images from an ND-IIITD retouched face dataset by utilizing pre-trained VGG16 TL models with ImageNet weights. Steps of the proposed method are pictured in Figure 2. An ND-IIITD retouched face images [24] dataset is used and we split the dataset into train and test (validation) sets of 80 % - 20 %, 70 % - 30 %, 60 % - 40 %, and 50 % - 50 %. Data transformation is applied with data augmentation. Two different optimizers with TL VGG16 are used during fine-tuning. Hence, training and evaluation are performed to these fine-tuned TL VGG16 models with different train-test split and optimizer sets. The test images are evaluated on all eight proposed models and the classification results are compared and analyzed. We evaluate these TL models, compare them, and suggest the best fine-tuned TL VGG16 model.

2.1. Fine-tuned and modified VGG16 architecture

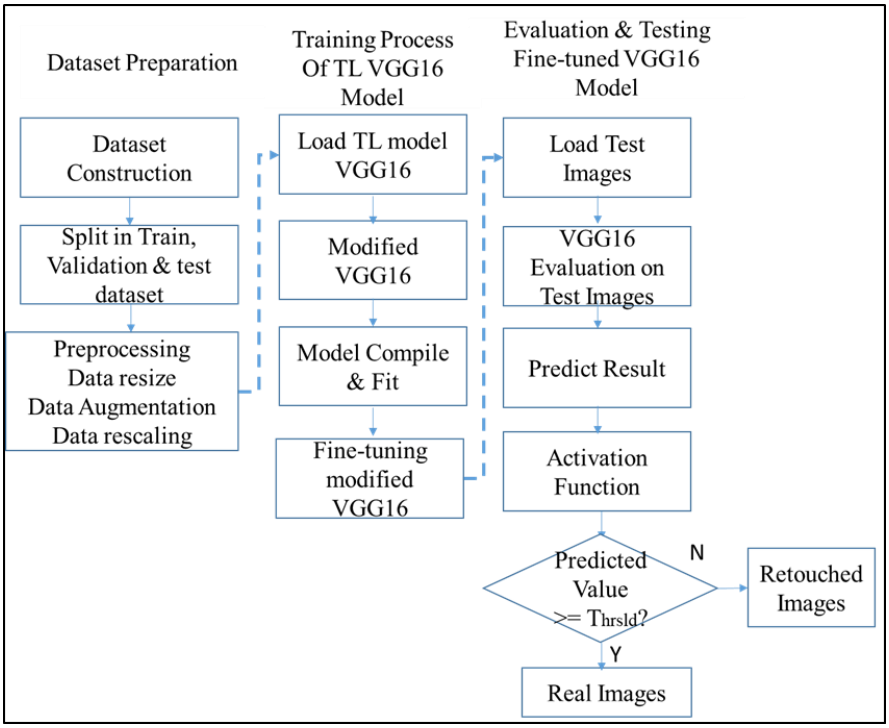


Figure 2. Steps for classification of retouching over ND-IIITD using TL fine-tuned VGG16 model.

The VGG16 model is a deep CNN that was presented by researchers in the Visual Geometry Group (VGG) at the University of Oxford [23]. It is a widely used model for image classification tasks and has achieved state-of-the-art performance on many benchmark datasets. The VGG16 model is a sequential architecture having 13 convolutional layers and 3 FC (fully connected) layers. The convolutional layers are used for extracting features of the input image,

while the FC layers perform the classification job. The architecture learns more complex features from the given input and keeps the parameters low. The FC layers of the original VGG16 are removed and a new FC layer is added for retouching classification. The information of trainable and non-trainable parameters before and after fine-tuning is given in Table 1.The architecture of the modified VGG16 is briefly described in Table 2. During initial training of the model all convolution layers are freeze and newly added FC layer is updated in terms of weight parameter. During fine-tuning, few convolution layers of the modified model are made unfreeze. Hence, the weights of unfreeze convolution layers and FC layer are updated.

2.2. Optimizers used during training

Table 1. Trainable parameters before and after fine-tuning.

Modified VGG16	Total parameters	Trainable parameters	Non-trainable parameters
Before fine-tuning	14,715,201	513	14,714,688
After fine-tuning	14,715,201	7,079,937	7,635,264

Deep learning optimizers are a crucial part of computer vision because they ensure that the training process produces the optimum outcomes. The optimizer's task is to minimize the loss, which gauges the discrepancy between expected and actual results, by repeatedly altering the model's parameters. The right optimizer can have a significant impact on the training's efficiency, speed, and accuracy, as well as the outcomes themselves [25]. As a result, optimizers are crucial in deep learning applications for computer vision.

Table 2. Modified VGG16 configuration.

Convolution layer No.	Channel	Filter size	Padding	Stride	O/P size	Max pooling size	Stride	O/P size
Input								224×224
1,2	64	3×3	1	1	224×224×64	2	2×2	112×112×64
3,4	128	3×3	-	-	112×112×128	2	2×2	56×56×128
5,6,7	256	3×3	-	-	56×56x256			28×28×256
8,9,10	512	3×3	-	-	28×28x512			14×14×512
11,12,13	512	3×3	-	-	14×14×512			7×7×512
Global average pooling								512
Drop out								512
FC								512
Output								1

2.2.1. Adam (adaptive moment estimation)

Adam is an optimization algorithm that combines the benefits of both RMSprop and momentum. Both the exponentially decaying average of the previous squared gradients (like

RMSprop) and the average of the past gradients (like momentum) are maintained. The name "Adam" is derived from "adaptive moment".

$$\boldsymbol{M_t = \beta_1 * M_{t-1} + (1 - \beta_1) * g} \quad (1)$$

$$\boldsymbol{V_t = \beta_2 * V_{t-1} + (1 - \beta_2) * g^2} \quad (2)$$

$$\widehat{\boldsymbol{M_t}} = \frac{\boldsymbol{M_t}}{\boldsymbol{1 - \beta_1^t}} \quad (3)$$

$$\widehat{\boldsymbol{V_t}} = \frac{\boldsymbol{V_t}}{\boldsymbol{1 - \beta_2^t}} \quad (4)$$

Update Rule,

$$\boldsymbol{\theta_{t+1} = \theta_t - \alpha * \widehat{M_t} \div \sqrt{\widehat{V_t}} + \epsilon} \quad (5)$$

where, $\boldsymbol{M_t}$ and $\boldsymbol{V_t}$ are the first and second moment estimates of the gradients, respectively; $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$ are hyperparameters that control the exponential decay rates of the moment estimates. $\boldsymbol{\alpha}$ is the learning rate, and $\boldsymbol{\epsilon}$ is a small value added for numerical stability.

2.2.2. RMSProp (root mean square propagation)

RMSprop is an optimizer that addresses the short comes of traditional stochastic gradient descent (SGD) by taking the learning rates of each parameter individually based on the historical average of the squared gradients.

$$\boldsymbol{v_t = \beta * v_{t-1} + (1 - \beta) * g^2} \quad (6)$$

$$\theta_{t+1} = \theta_t - \alpha * g \div \sqrt{v_t} + \epsilon \quad (7)$$

where, $\boldsymbol{v_t}$ is exponential decaying average of squared gradients; $\boldsymbol{\beta}$ is a hyper parameter that controls the exponential decay rates of the moment estimates; $\boldsymbol{\alpha}$ is the learning rate, and $\boldsymbol{\epsilon}$ is a small value added for numerical stability.

2.3. ND-IIITD retouched faces dataset

For the retouching detection, the real and retouched images of ND-IIITD retouched faces are taken. The ND-IIITD dataset [26] contains real (bonafide) and retouched images of a total of 325 subjects, where 211 are male faces and 114 are female faces. Each subject is further divided into 7 real face images taken during different time spans or under different light conditions, different backgrounds and with different poses. Those 7 real probes are retouched or altered with Portrait ProMax, a photo editing tool, with different levels of retouching. Retouching or brushing is applied on the nose, eye, lips, chick, and hair areas of the bonafide images [6].

Table 3. Dataset description [Pr: degree/percentage of the retouching. Pr 1: least alteration and Pr 7: max alteration].

ND-IIITD retouched faces	Bonafide images	Retouched Images
Pr 1	325	325
Pr 2	325	325
Pr 3	325	325
Pr 4	325	325
Pr 5	325	325
Pr 6	325	325
Pr 7	325	325
Total	2600	2275

Table 4. Train-test splitting of bonafide and retouched images.

Training-testing split ratio	Train dataset	Validation dataset	Test dataset
	No. of images (bonafide + retouched)	No. of images (bonafide + retouched)	No. of images (bonafide + retouched)
80 % - 20 %	3624	462	460
70 % - 30 %	3175	686	684
60 % - 40 %	2713	910	920
50 % - 50 %	2267	1133	1146

Table 3 shows the description of the original ND-IIITD dataset and Figure 3 shows some samples of retouching. The accuracy of the model depends on several parameters such as learning rate, number of epochs, optimizers, data size, etc. The ND-IIITD dataset is divided into different train test split ratios and the performance of all is compared for maximum classification accuracy. The details of bifurcation of dataset into training and testing sets are described in Table 4. The ratio of 80 % - 20 % means that 80 % of the images are considered for the train dataset and 20 % are equally divided into the validation and test datasets. The model is trained on the training dataset and evaluated and tested on the testing dataset. Bonafide (real) and retouched samples of ~105 male and ~57 female subjects are used for training and the rest half (i.e. ~105 males and ~57 females) are used for evaluation. Each dataset is formed by having the same number of real samples (images) as that of retouched samples.

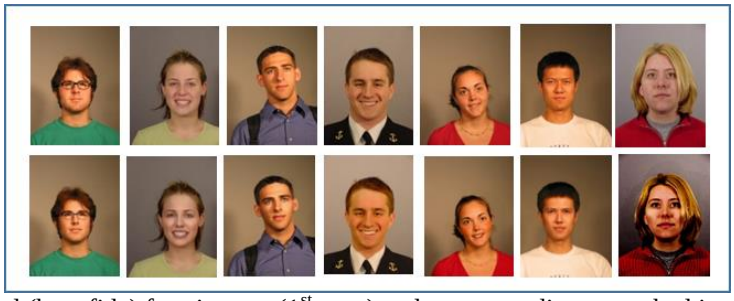


Figure 3. Real (bonafide) face images (1st row) and corresponding retouched images from preset 1 to 7 (2nd row) [24].

3. RESULTS AND DISCUSSION

3.1. Experimental arrangement

The model training and evaluation tasks are conducted on Google Colab with GPU runtime, utilizing TensorFlow, a machine learning package developed by Facebook's AI Research Department. A Python data loader is employed to load the data with a batch size of 32, and Google Drive serves as the storage location for both the dataset's file names and checkpoints. All the graphs are plotted using Matplotlib library. For the initial training of the modified fine-tuned VGG16 model, we use the Adam optimizer with a learning rate (LR) of 0.001, β_1 and β_2 of 0.9 and 0.999, respectively, and a number of epochs set to 10. During the fine-tuning of the model, we use the Adam and the RMSprop optimizers (momentum 0) with an LR of 0.0001 and a number of epochs set to 20. With the above hyper parameter settings, a total of eight experiments are conducted. The modified VGG16 model is trained over a training (train) dataset and evaluated over a testing (validation) dataset. On the train and validation sets, we determine the cross-entropy loss and accuracy for each epoch.

3.2. Performance metrics for evaluation

For the classification task, the performance of the model is evaluated based on 4 different parameters, namely precision, sensitivity, F1-score, and accuracy for bonafide and retouched face images. The ability of the models to give maximum TPR (true positive rate) is compared by the ROC (region of curve).

Precision (P) is a metric that defines the percentage of correctly calculated results and is stated as:

$$P = \frac{x}{x+y} \tag{8}$$

where, *x*: true positive samples and *y*: false positive samples.

Recall (R) can be represented as the ratio of TP (true positives) to the sum of TP (true positives) and FN (false negatives), and expressed as:

$$R = \frac{x}{x+z} \tag{9}$$

where, *x*: true positive samples and *z*: false negative samples.

F1-score (F1) analyses a model's performance on a class-by-class basis to determine how predictive it is.

$$F1 = 2 * \frac{[P*R]}{[P+R]} \tag{10}$$

Accuracy (A) is calculated as the proportion of accurately identified forecasts to all predictions.

$$A = \frac{correct\ predictions}{total\ predictions} \tag{11}$$

ROC (region of curve) is a graphical measure of diagnostic ability of the model. It is a graphical plotting of TPR (true positive rate) vs. FPR (false positive rate) for the proposed model.

3.3. Result analysis based on training and validation accuracy and cross entropy

Table 5. Comparison of accuracy and cross entropy for training dataset.

Train-test split	Optimizer used after fine-tuning	Accuracy before fine-tuning	Accuracy after fine-tuning	Loss before fine-tuning	Loss after fine-tuning
80 % - 20 %	Adam	0.6929	0.9928	0.5548	0.0211
70 % - 30 %		0.6918	0.9918	0.5550	0.0241
60 % - 40 %		0.7030	0.9934	0.5608	0.0175
50 % - 50 %		0.6873	0.9894	0.5649	0.0299
80 % - 20 %	RMSprop	0.6970	0.9832	0.5546	0.0545
70 % - 30 %		0.6905	0.9852	0.5590	0.0663
60 % - 40 %		0.6828	0.9834	0.5703	0.0598
50 % - 50 %		0.6846	0.9815	0.5718	0.1104

As per Table 5, when Adam optimizer is used during fine-tuning, the accuracy achieved for all train-test split ratios is ~ 99 % as compared to the cases with RMSprop optimizers. Again, for

the 50 - 50 % train-test split, the model accuracy is incremented by 30.21 % over 30 epochs. The cross-entropy is reduced by ~1 to 2 % for all train-test splits with the Adam optimizer used during fine-tuning.

Table 6. Comparison of accuracy and cross entropy for testing (validation) dataset.

Train-test split	Optimizer used after fine-tuning	Accuracy before fine-tuning	Accuracy after fine-tuning	Loss before fine-tuning	Loss after fine-tuning
80 % - 20 %	Adam	0.6602	0.9545	0.5297	0.2238
70 % - 30 %		0.7187	0.9548	0.5172	0.2154
60 % - 40 %		**0.7330**	**0.9791**	0.5396	**0.0938**
50 % - 50 %		**0.7087**	**0.9744**	0.5626	**0.1113**
80 % - 20 %	RMSprop	0.6645	0.9784	0.5194	0.1653
70 % - 30 %		0.7201	0.9359	0.5222	0.2424
60 % - 40 %		0.7286	0.9495	0.5517	0.2457
50 % - 50 %		0.6708	0.9709	0.5672	0.1220

Table 7. Comparison of training time for different train-test split ratios.

Train-test split ratio	Optimizer used during fine-tuning	Wall time (min)		
		Initial training	Fine-tuning	Total
80 % - 20 %	Adam	12.1	26.26	38.27
70 % - 30 %		11.9	25.33	36.42
60 % - 40 %		11.24	24.35	35.59
50 % - 50 %		12.13	24.42	36.52
80 % - 20 %	RMSprop	13.1	26.31	39.32
70 % - 30 %		12.53	25.26	38.19
60 % - 40 %		11.15	24.18	35.33
50 % - 50 %		10.42	20.30	31.12

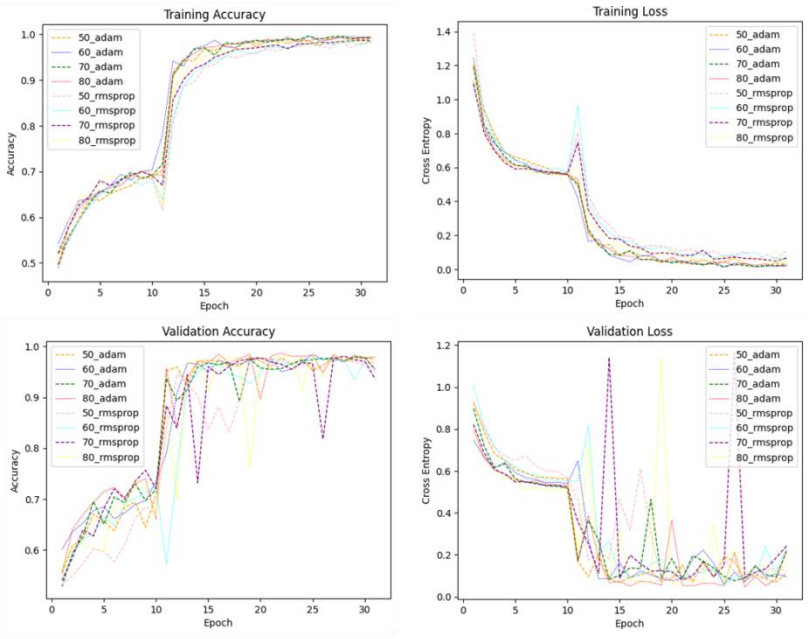


Figure 4. Comparison of all suggested TL fine-tuned VGG16 models.

According to Table 6, maximum accuracy and minimum cross entropy loss are achieved for Adam optimizers and 60 - 40 % train-test split. Although, 50 - 50 % split with Adam gives a nearly equal performance in terms of accuracy and loss. The epoch-wise comparison for all 8 experiments is depicted in Figure 4, revealing that the model's accuracy starts to improve after epoch 10, coinciding with the commencement of fine-tuning.

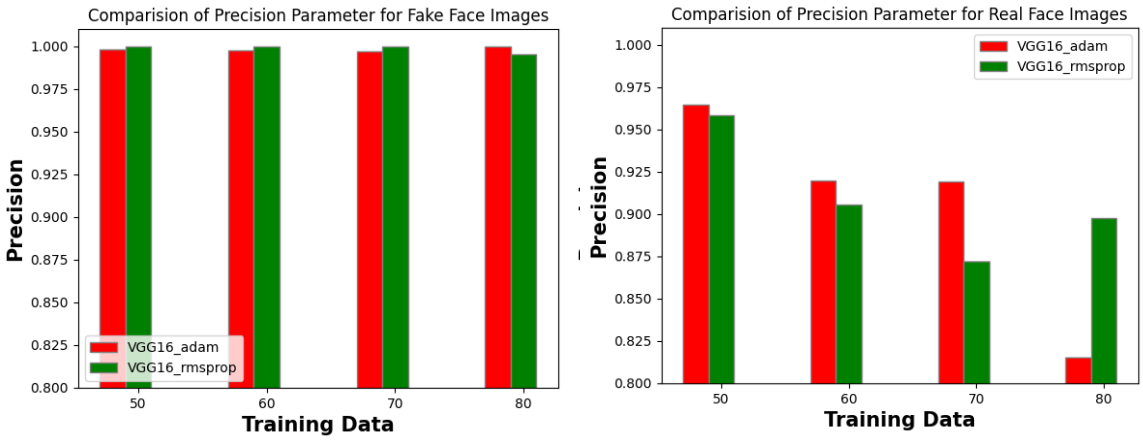
3.4. Result analysis based on training time

The information mentioned in Table 7 is presented by running %time of python before every training of the model. This returns the wall time or clock time required to train the model over the ND-IIITD dataset. Toughly, these timings depend on various factors like CPU usage, and the random samples taken by the model during batch normalization. Hence, a proper comparison over optimizers and the split ratios is not possible. In our analysis, we focus on training timing parameters that have not been previously explored or mentioned in the papers discussed in Subsection 1.2. We aim to uncover new insights and potential optimizations that could improve the efficiency and speed of the training process, providing a novel contribution to the existing methods.

3.5. Result analysis based on performance metrics

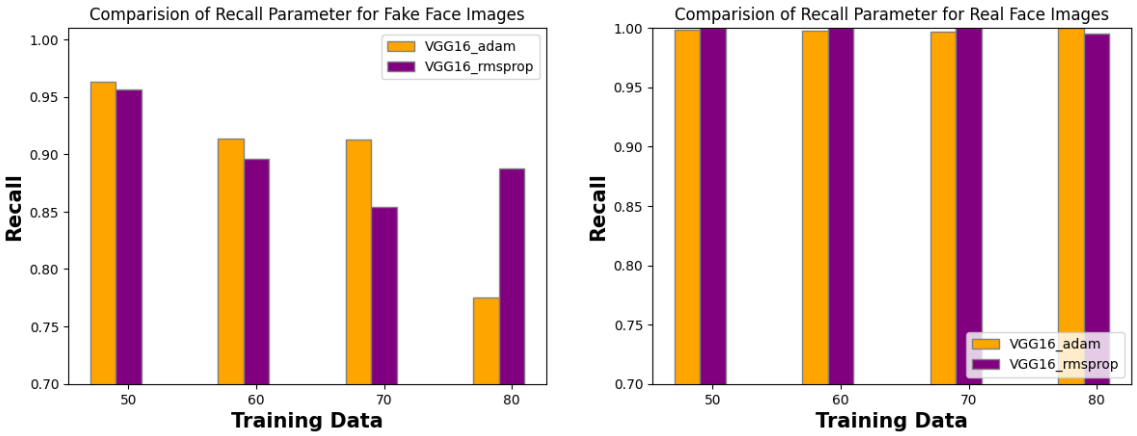
As per Figures 5.1(a) & (b), Adam and RMSprop optimizers used during fine-tuning perform equally with ~100 % precision for retouched (fake) images. On the other side, for 70 %, 60 % and 50 % splitting, Adam optimizer gives better results in terms of precision parameters for bonafide (real) images. Out of all split ratios, for 50 % train-test split, the maximum precision for real samples achieved is 96.45 % and 95.81 % when Adam and RMSprop are used, respectively.

As shown from Figures 5.2(a) & (b), out of all split ratios, 50 % ratio gives better accuracy in terms of recall metric for classifying retouched face images. The recall values measured by proposed fine-tuned VGG16 model with Adam optimizer are 96.34 % (max), 91.34 %, 91.25 % and 77.49 % for 50 %, 60 %, 70 % and 80 % split ratios, respectively. And, they are 95.64 % (max), 89.61 %, 85.42 % and 88.74 % for that of RMSprop optimizer. For classifying real images, the proposed TL VGG16 gives an accuracy of ~100 % for all split ratios with both the



(a) (b)

Figure 5.1. Comparison of precision parameter: (a) Retouched(fake) face samples; (b) Bonafide face samples.



(a) (b)

Figure 5.2. Comparison of recall parameter: (a)Retouched face samples; (b) Bonafide face samples.

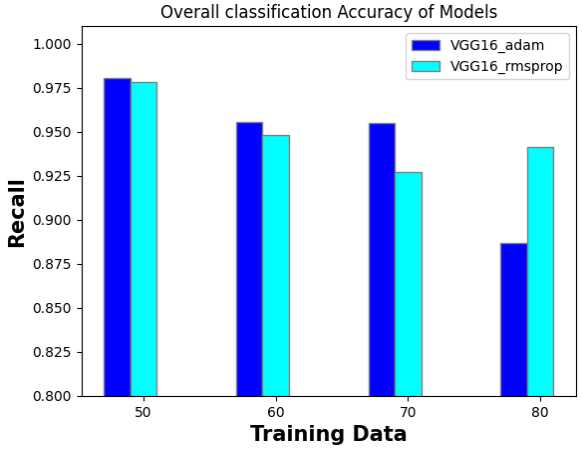


Figure 5.3. Accuracy analysis.

optimizers as shown in Figure 5.2(b). As depicted in Figure 5.3, over all split ratios, for the Adam and RMSprop optimizers used during fine-tuning of the proposed TL VGG16 model, the maximum accuracy achieved is 98.08 % and 97.82 %, respectively, for 50 % - 25 % of train-test split ratio.

3.6. Comparison with existing works

The proposed model achieved the highest overall classification accuracy of 98.08 %, and the classification accuracy of retouched and real images of 99.83 % and 96.344 %, respectively. Moreover, the proposed model shows improvement of 16.18 % and 10.98 % in classifications of retouching for the same dataset, as shown in Table 8. Even ROC comparison of the proposed work reflected in Figure 5.4 with [6] demonstrates that the suggested model indicates better overall performance in terms of true positive rate versus false positive rate.

As compared to Reference [13], recent paper, the proposed work improved the classification accuracy by 10.08 % when the model is trained and evaluated on the whole image

rather than the face patches. These findings lead to the conclusion that the proposed model exhibits superior performance in discerning genuine from retouched images when compared to state-of-the-art models. Most prior studies on retouching have trained and evaluated models using facial patches defined by specific landmarks, which often do not achieve optimal accuracy when analyzing entire images. In contrast, our research demonstrates enhanced accuracy and classification performance when training the model on entire images.

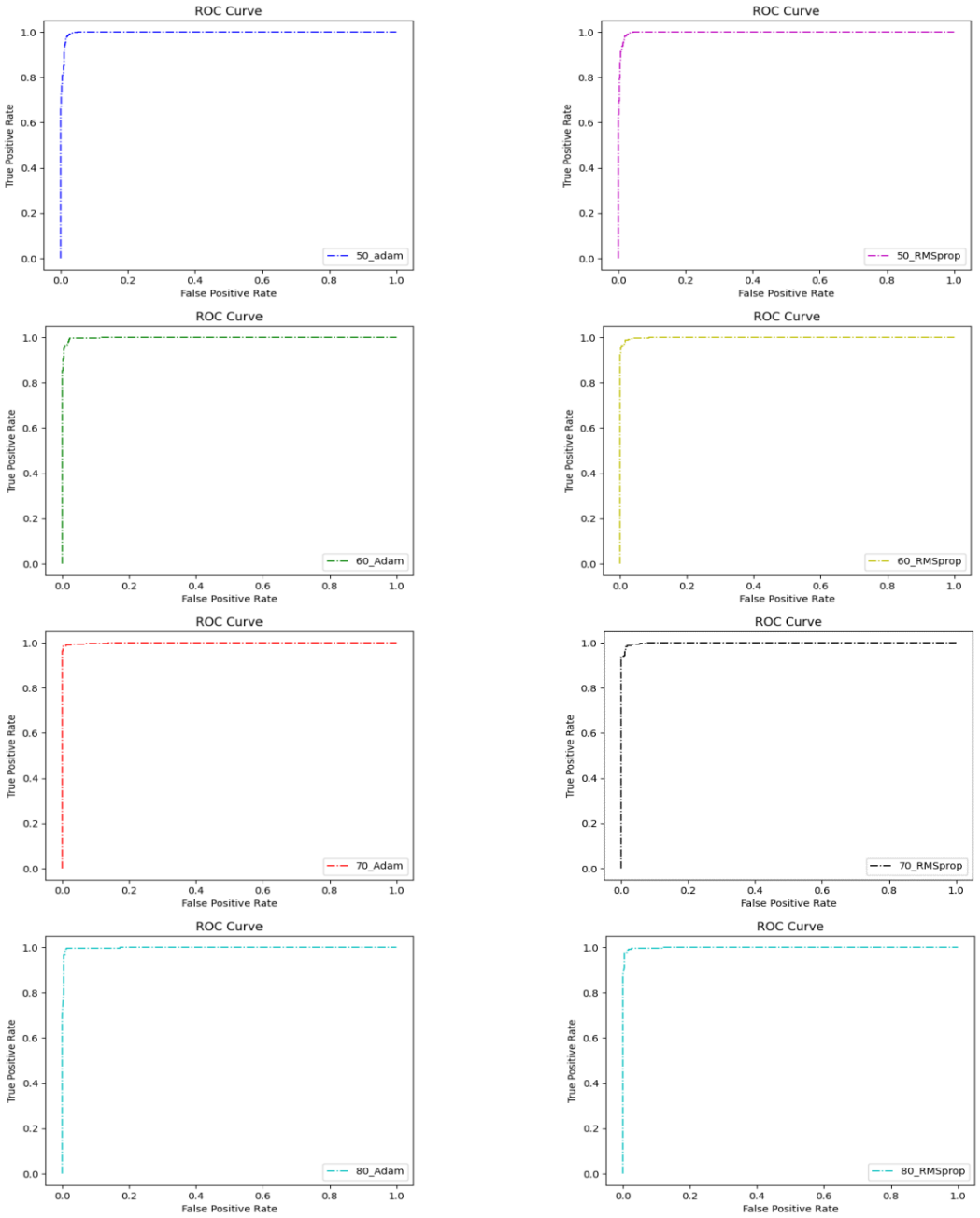


Figure 5.4. Analysis of curve region.

Table 8. Comparison of proposed model with all existing models.

Reference	Method	Train-test split	Overall	Bonafide	Retouched
Kee and Farid [4]	-	-	48.80 %	32.70 %	71.90 %
Bharti *et al*. [6]	Unsupervised DBM	50 % - 50 %	81.90 %	74.30 %	90.90 %
	Supervised DBM		87.10 %	81.10 %	93.90 %
Sharma *et al*. [13]	Residual CNN		90.00 %	93.30 %	86.30 %
Proposed fine-tuned VGG16 model	Adam optimizer	50 % - 50 %	98.08 %	99.83 %	96.34 %
	RMSprop optimizer		97.82 %	100 %	95.64 %

4. CONCLUSIONS

This paper presents a transfer learning approach to detect the digital manipulation of facial images. The pre-trained VGG16 model gives better performance over a small dataset compared to existing methods. Furthermore, leveraging the transfer learning and fine-tuning reduces the computational complexity and time. The experimental results demonstrate the significance of choosing appropriate data partitioning and optimization techniques in enhancing the overall performance of the VGG16-based classifier for retouched face image recognition tasks. The work shows that the fine-tuned VGG16 model with Adam optimizer outperforms to classify real and retouched faces for 50 % - 25 % train-test split ratio over the ND-IIIITD retouched face dataset.

In the future, we can explore the use of other facial datasets and incorporate more pre-trained models to investigate proper data partitioning and optimization techniques. By experimenting with different datasets and optimizers, we can potentially enhance the fine-tuned VGG16 model's performance in facial retouching detection. Additionally, further research on data augmentation and fine-tuning strategies may help improve the generalization and robustness of the model for real-world applications.

Acknowledgements. We are thankful to the University of Notre Dame for providing the ND-IIITD retouched face dataset for our research work.

CRediT authorship contribution statement. Prof. Kinjal R Sheth: Methodology, Implementation, Investigation, and Writing. Dr. Vishal S Vora: Supervision and Review.

Declaration of competing interest. We have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

1. Russello S. - The Impact of Media Exposure on Self-Esteem and Body Satisfaction in Men and Women, Vol. 1, 2009.
2. Gupta S. - JIPR **10** (6) (2005) 491-498.
3. Altabe M. - Ethnicity and body image: Quantitative and qualitative analysis, Int. J. Eat. Disord. **23** (2) (1998) 153-159. doi: 10.1002/(SICI)1098-108X(199803)23:2<153::AID-EAT5>3.0.CO;2-J.
4. Kee E. and Farid H. - A perceptual metric for photo retouching, Proc. Natl. Acad. Sci. U.

S. A. **108** (50) (2011) 19907-19912. doi: 10.1073/pnas.1110747108.

5. Kose N., Apvrille L., and Dugelay J. L. - Facial makeup detection technique based on texture and shape analysis, 2015 11th IEEE Int. Conf. Work. Autom. Face Gesture Recognition, FG 2015, 2015, doi: 10.1109/FG.2015.7163104.
6. Bharati A., Singh R., Vatsa M., and Bowyer K. W. - Detecting Facial Retouching Using Supervised Deep Learning, IEEE Trans. Inf. Forensics Secur. **11** (9) (2016) 1903-1913. doi: 10.1109/TIFS.2016.2561898.
7. Bharati A., Vatsa M., Singh R., Bowyer K. W., and Tong X. - Demography-based facial retouching detection using subclass supervised sparse autoencoder, arXiv, 2017.
8. Singh A., Tiwari S., and Singh S. K. - Face tampering detection from single face image using gradient method, Int. J. Secur. its Appl. **7** (1) (2013) 17-30.
9. Rathgeb C., *et al.* - PRNU-based detection of facial retouching ISSN 2047-4938, IET Biometrics **9** (4) (2020) 154-164. doi: 10.1049/iet-bmt.2019.0196.
10. Seibold C., Hilsmann A., and Eisert P. - Reflection Analysis for Face Morphing Attack Detection, 2018 26th Eur. Signal Process. Conf., 2018, pp. 1022-1026.
11. Ciftci U. A. - FakeCatcher : Detection of Synthetic Portrait Videos using Biological Signals, Vol. X, no. X, 2020, pp. 1-17.
12. Akhtar Z., Dasgupta D., and Banerjee B. - Face Authenticity : An Overview of Face Manipulation Generation , Detection Available on : Elsevier-SSRN Face Authenticity : An Overview of Face Manipulation Generation , Detection and Recognition, May, 2019.
13. Sharma K., Singh G., and Goyal P. - IPDCN2 : Improvised Patch-based Deep CNN for facial retouching detection, Vol. 211, May 2021, 2023.
14. Alzubaidi L., *et al.* - Review of deep learning : concepts , CNN architectures , challenges, applications , future directions, Springer International Publishing, 2021.
15. Krishna S. T. and Kalluri H. K. - Deep learning and transfer learning approaches for image classification, Int. J. Recent Technol. Eng. **7** (5) (2019) 427-432.
16. Desai C. G. and N. Academy D. - Image Classification Using Transfer Learning and Deep Learning, September 2021. doi: 10.18535/ijecs/v10i9.4622.
17. Ying Q., Liu J., Li S., Xu H., Qian Z., and Zhang X. - RetouchingFFHQ : A Large-scale Dataset for Fine-grained Face Retouching Detection."
18. Bichri H., Chergui A., and Hain M. - ScienceDirect Image Classification with Transfer Learning Using a Custom Dataset : Comparative Study Image Classification with Transfer Learning Using a Custom Dataset : Comparative Study, Procedia Comput. Sci. **220** (2023) 48-54. doi: 10.1016/j.procs.2023.03.009.
19. Ibrahim A. M., Elbasheir M., Badawi S., Mohammed A., and Alalmin A. F. M. - Skin Cancer Classification Using Transfer Learning by VGG16 Architecture (Case Study on Kaggle Dataset), 2023, pp. 67-75. doi: 10.4236/jilsa.2023.153005.
20. Sheth K. R. and Vora V. S. - A comparative study on image forgery-facial retouching **12** (2) (2023) 851-859. doi: 10.11591/eei.v12i2.4481.
21. Shyu M., Chen S., and Iyengar S. S. - A Survey on Deep Learning : Algorithms, Techniques, ACM Comput. Surv. **51** (5) (2018) 1-36.
22. Sharma N. and Sharma N. - An Neural An Analysis Analysis Of Of Convolutional

Convolutional Neural Networks Networks For For Image Image An Analysis of Co Classification an Analysis of Convolutional Neural Networks for Image Classification an Analysis of Convolutional Neural and Ne, Procedia Comput. Sci. **132** (Iccids) (2018) 377-384. doi: 10.1016/j.procs.2018.05.198.

23. Simonyan K. and Zisserman A. - Very deep convolutional networks for large-scale image recognition, 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., 2015, pp. 1-14.
24. face Database N. I. R. - https://cvrl.nd.edu/projects/data/. Please replace by other suitable ref., for example, journal, book, proceeding, etc.
25. Kandel I. and Castelli M. - Comparative Study of First Order Optimizers for Image Classification Using Convolutional Neural Networks on Histopathology Images, 2020.
26. Jain A., Singh R., and Vatsa M. - On detecting GANs and retouching based synthetic alterations, 2018. doi: 10.1109/BTAS.2018.8698545.