An intelligent approach to detect facial retouching using Fine Tuned VGG16

Kinjal Ravi Sheth, Dr. Vishal S. Vora

Department of Electronics and Communication Engineering,
Faculty of Engineering and Technology,
Atmiya University, India
Email: kinjal.ddit@gmail.com, vishal.vora@atmiyauni.ac.in

Abstract: It is a common practice to digitally edit or 'retouch' facial images for various purposes, such as enhancing one's appearance on social media, matrimonial sites, or even as an authentic proof. When regulations are not strictly enforced, it becomes easy to manipulate digital data, as editing tools are readily available. In this paper, we apply a transfer learning approach by fine-tuning a pre-trained VGG16 model with ImageNet weight to classify the retouched face images of standard ND-IIITD faces dataset. Furthermore, this study places a strong emphasis on the selection of optimisers employed during both the training and fine-tuning stages of the model to achieve quicker convergence and enhanced overall performance. Our work achieves impressive results, with a training accuracy of 99.54% and a validation accuracy of 98.98% for the TL vgg16 and RMSprop optimiser. Moreover, it attains an overall accuracy of 97.92% in the two-class (real and retouching) classification for the ND-IIITD dataset.

Keywords: Adam; retouching; RMSprop; transfer learning; TL; VGG16.

Reference to this paper should be made as follows: Sheth, K.R. (2024) 'An intelligent approach to detect facial retouching using Fine Tuned VGG16', *Int. J. Biometrics*, Vol. 16, No. 6, pp.583–600.

Biographical notes: Kinjal Ravi Sheth is an Assistant Professor at L.D. College of Engineering, Ahmedabad, Gujarat, India since 2011. She is also a PhD Research Scholar of Atmiya University at Department of Electronics and Communication, Rajkot, Gujarat, India. She has completed her Master of Engineering from Dharmasinh Desai university, Nadiad in 2008.

1 Introduction

Face recognition has been a very busy study area over the previous few decades (Abate et al., 2007; Parkhi et al., 2015; Jain, 2014). Deep convolutional neural networks have recently demonstrated substantial performance increases in facial recognition systems, image classification problems and many more. However, a variety of factors, such as differences in pose and gesture, facial expression, or quality of image, have been identified that can reduce the recognition accuracy of the recognition system (Shyu et al., 2018). Moreover, it was shown that face beautification catches digitally, also known as facial retouching, has the capacity to drastically modify how a human face is viewed in terms of shape and texture (Rathgeb et al., 2019). Similar changes to those produced by

plastic surgery (Singh et al., 2010) or face cosmetics (Dantcheva et al., 2012) are brought about by facial retouching. In the digital world, it is possible to make additional cosmetic changes to face images, such as enlarging or repositioning the eyes, modifying the shape of the lips and chaw, altering the region of the forehead, etc. A number of mobile apps, in addition to professional photo editing programs like Photoshop, provide a variety of filters and other effects that even inexperienced users may easily utilise. An example of facial retouching using a well-known beauty app is shown in Figure 1.

Figure 1 Beauty enhancement of a beautification app, (a) 1st row contains original (real) images (b) 2nd row contains retouched images using photo editing software (see online version for colours)

Source: Images taken from 57 Celebrities Before And After Photoshop Who Set Unrealistic Beauty Standards (2017)

Advertising or magazine that features digitally altered faces of model or celebrities frequently has an impact on consumer behaviour. These digitally enhanced photos should be identified accurately to avoid any kind of dangerous consequences or growing risk of disordered (Antony, 2021). In response, so many country like Israel, France, Italy passes a law to lessen the growing risk of disorders (Photoshop, 2022; Eggert, 2017; Smith, 2022). Hence, to increase the face appeal, minor editing, such as skin smoothing, blemish removal, and hair colour changes has been done and the retouched photos are often uploaded on social media and as an identity proof (Gupta, 2005). In this context, the work carried out in this paper showcases the efficiency of detecting retouching done over face images using photo editing tools.

1.1 Research gap

The introduction section discusses how the retouching or spoofing attack degrade the performance of any face recognition system. The major points of discussion of the research work carried out till for classifying retouched face images are:

1 To train the model very huge amount of facial data (real + retouched) is required.

2 To train such huge data, the convergence time is too high.

3 The hardware requirement to cope with such huge data is again very high.

4 The iteration or epoch required to train the model on training and validation dataset are required to set around multiple of hundreds to achieve the better accuracy.

The mentioned 4 points actually affect the efficiency of the model and the timing requirement. Hence, the main aim of this research is that above mention drawbacks should be overcome by introducing transfer learning (TL) approach. Our work makes the following contributions:

- We propose the utilisation of a fine-tuned VGG16 model, employing TL with pre-trained weights from 'ImageNet', to effectively classify real and retouched images in the ND-IITD facial retouching dataset.
- In order to address the classification problem, we enhance the VGG16 model by incorporating an additional fully connected (FC) layer beyond the default architecture, allowing for fine-tuning. This ultimately reduced the computation parameters as compared to basic VGG16 model.
- In the literature, either a classifier or the Adam optimiser was employed for expediting model convergence during training. To showcase the effectiveness of our proposed TL model, we conducted four experiments using the ND-IIITD retouched face dataset. During both the initial training and fine-tuning phases, we utilised four pairs of optimisers and compared our results with existing models. Specifically, we employed the Adam and RMSprop optimisers and conducted a comparative analysis of the outcomes to assess the model's performance under varying optimisation strategies.

The content of this research is organised as follows: A literature review is presented in Section 2. Afterwards, the brief of TL approach, VGG16 architecture, data augmentation, optimisers and facial dataset used for proposed methodology are outlined in Section 3. Implementation of the proposed TL retouching detection methodology is introduced in Section 4. Classification results are summarised in Section 5 and conclusion is discussed in Section 6.

1.2 Literature review

Facial spoofing, morphing on face images, make-up detection is considered under the attack of retouching. The relevant works detecting the facial retouching along with the datasets used and applied method to classify the retouched images are discussed in Gupta (2005). Kee and Farid (2011) had trained a support vector regression for the fake and real images. For training the SVR on different celebrity images geometric as well as

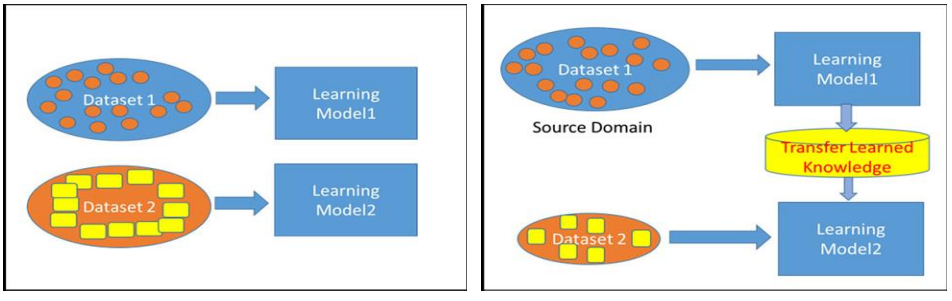
photometric features are used. Research work carried out by Bharati et al. (2016) used supervised deep learning method for classifying retouching on the ND-IIITD facial retouched database. They had presented their new ND-IIITD dataset which includes of 2,600 original and 2,275 doctored images. In 2017, Bharati et al. (2017) developed an approach to report the retouching accuracy on the multi-demographic retouched faces (MDRF) dataset using semi-supervised autoencoders. The article by Kose et al. (2015) employs SVM classifier on a features vector comprising of features of shape and texture in the broader field of face image forensics. The accuracy of makeup detection is measured on YMC and VMU datasets. Singh et al. (2013) proposed an algorithm which classify tampered face images. The author uses gradient based approach for classification. Jain et al. (2018) proposed a supervised DL approach using CNNs to detect and classify fake images. The authors used ND-IIITD retouched database for classify the altered images. As a 2nd experiment, they have evaluated performance accuracy on real celebrities database – CelebA and StarGAN (Choi et al., 2018) generated fake celebrities images. Rathgeb et al. (2020) presents retouching detection using a PRNU (photo response non-uniformity) analysis. Five apps after qualitative assessment are selected to create a database of 800 face images. Above mentioned existing algorithms uses machine learning and deep learning approach to classify the retouching. The patch based deep CNN architecture is used to classify real and retouched images (Sharma et al., 2022). In the pre-processing stage, 68 facial landmarks were utilised to extract pertinent patches from the input image. The second stage employs an efficient and resilient CNN based on residual learning to extract high-level hierarchical features from these patches. The classification accuracy achieved is 99.84% on ND-IIITD dataset but when the model is tested over whole images rather than patches the accuracy achieved is 90% only.

Based on our literature survey and as per best of our knowledge, there is no evidence of anyone leveraging TL knowledge for retouching detection. In response, we propose a modified VGG16 model that incorporates TL, enhancing its ability to accurately classify real and retouched face images. This modification involves the addition of just one FC layer on top of the standard VGG16 model.

2 Transfer learning

Using a sizable collection of labelled images, deep learning algorithms are often employed to train a model from scratch. This entails creating the architecture of a neural network, initialising its weights arbitrarily, then improving these weights via backpropagation to reduce a loss function. The resultant model could be used to classify new test images into one of the pre-determined categories. Contrarily, TL is modifying a previously trained neural network model for a new classification problem using input from a smaller dataset. The pre-trained model has already mastered the art of extracting features from images, and one can use these features as the basis for a new model. Depending on how closely the target dataset resembles the pre-trained dataset, the weights of the pre-trained model can either be left unchanged or adjusted on the new dataset (Hussain et al., 2019; Ramdan et al., 2020; Cyriac et al., 2021).

Figure 2 (a) Traditional learning approach, (b) transfer learning approach (see online version for colours)

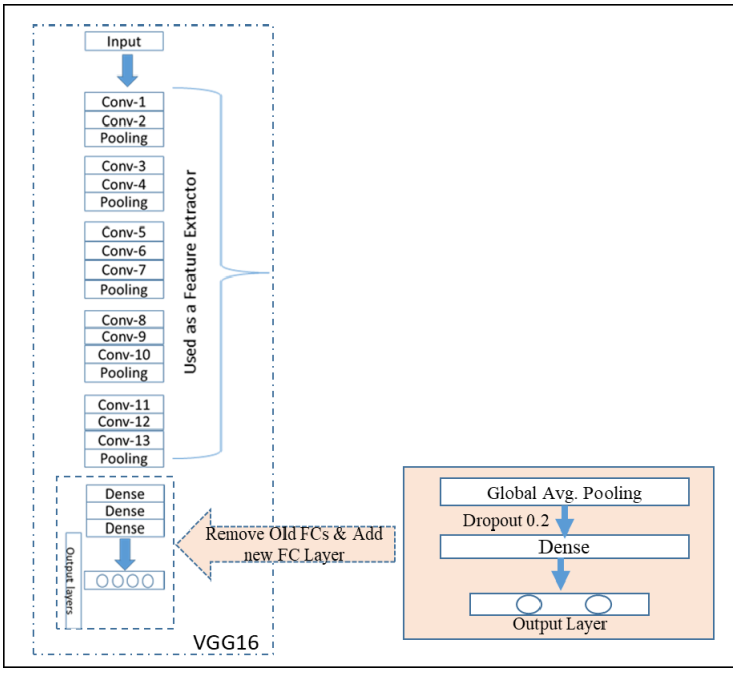


(a) (b)

2.1 VGG16 architecture

The visual geometry group (VGG) at the University of Oxford introduced the VGG16 model (Simonyan and Zisserman, 2015) in 2014, which is a deep convolutional neural network. The VGG16 model has undergone testing across a range of publicly available, extensive datasets. Specifically, when applied to image classification tasks on the 'ImageNet' dataset, it often demonstrates top-5 validation error rates of around 7.5% and a test error rate of approximately 7.3%. These performance metrics establish the model as a promising candidate for our research in the realm of retouching classification tasks.

Figure 3 Vgg16 model and architecture (see online version for colours)



Note: the modification is shown by pink rectangle

The VGG16 model consists of 16 layers, contains thirteen convolutional layers and three FC layers. Accepts input images of fixed size (commonly 224 × 224 pixels). Consists of 13 convolutional layers, where each layer uses a small 3 × 3 filter. The convolutional layers are followed by rectified linear unit (ReLU) activation functions, which introduce non-linearity. After every two consecutive convolutional layers, there is a max-pooling layer with a 2 × 2 window and a stride of 2. Max-pooling reduces the spatial dimensions of the feature maps and helps in translation invariance. There are three FC (dense) layers towards the end of the network. The first two FC layers have 4,096 neurons each, followed by ReLU activation functions. The final FC layer has 1,000 neurons, corresponding to the 1,000 classes in the ImageNet dataset, and it uses a softmax activation function for classification. The output layer produces probabilities for each of the 1,000 classes in the ImageNet dataset.

The top FC layers are first removed, followed by the application of global average pooling. Subsequently, a dropout rate of 0.2 is applied which is applied to learn the model more generalise over features rather than overfitting. Finally, a single FC layer with 513 input neurons and 2 output neurons is added in the output layer. This modification streamlines the model's architecture, making it well-suited for the binary classification task of distinguishing between genuine and retouched facial images, while also reducing computational complexity.

2.2 Optimisers

Optimisers are algorithms used in DL to optimise the weights and biases of a neural network during the training process. The objective is to reduce a loss function, which gauges the discrepancy between the network's predicted and actual output. The quality and speed of convergence during training can be significantly impacted by the optimiser selection. There are several different types of optimisers, like, Gradient descent, Stochastic gradient descent, Adaguard, RMSprop, Adadelta, Adamax, etc., each with its own specific approach for updating the parameters. 'A comparative analysis of different optimisers on histopathology images' (Kandel and Castelli, 2020) showcased the comparison of different optimisers for the histopathology datasets. Inspired from the comparison presented, this study uses two widely used optimiser for classification problems, RMSprop and Adam. The effect of these two on the face classification is analysed and compared to consider best fit model.

2.2.1 RMSprop

Root Mean Square Propagation, is an adaptive learning rate optimisation algorithm that divides the learning rate by an exponentially decaying average of squared gradients (Gupta, 2021). This helps to scale down the learning rate for features with high gradients and scale up the learning rate for features with low gradients. As a result, the algorithm can converge faster and be less prone to getting stuck in local minima. The update rule for RMSprop is as follows:

$$w_t^i = w_{t-1}^i - \frac{\eta}{\sqrt{E[G^2]t+\in}} \square\nabla_w C\left(w_t^i\right) \tag{1}$$

$$E[G^2]_t = \lambda E[G^2]_{t-1} + (1-\lambda)G_t^2 \tag{2}$$

$$G = \nabla_w C(w_t) \tag{3}$$

where

$C(.)$ cost function

G the gradient of parameters w_t for (x image, y label)

η learning rate hyper parameter

λ select the amount of information of the previous update

$E[G^2]$ running avg. of the squared gradient.

2.2.2 Adam (adaptive moment estimation)

Adam is another gradient descent algorithm that combines the benefits of both RMSprop and momentum optimisation (Li et al., 2019). It maintains a moving average of both the gradient and its squared gradient, and uses them to update the parameters during training. Adam is known for its excellent performance on a wide range of deep learning tasks, including computer vision and natural language processing. The weight is updates as follows (Gupta, 2021):

$$w_t^i = w_{t-1}^i - \frac{\eta}{\sqrt{\hat{v}_t + \in}} \cdot \hat{m}_t \tag{4}$$

where

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{5}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{6}$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) G \tag{7}$$

$$v_t = \beta_2 v_{t-1} (1 - \beta_2)[G]^2 \tag{8}$$

$$G = \nabla_w C(w_t) \tag{9}$$

where

$C(.)$ the cost function

G gradient of w_t for (x image, y label)

η learning rate

$\nabla_w C(w_t)$ is the gradient of weight parameters w_t for image x and its corresponding label y,

$E[G^2]$ the running average of the squared gradients

β_i the first moment

v_t running average to select the amount of information from the past update, where β_i between [0, 1]

m_t the squared gradients.

2.3 Loss function

In machine learning, a loss function is a math function that measures the difference between the predicted and actual output of a model. A machine learning model is trained with the intention of minimising the loss function, which means finding the set of model parameters that produce the most accurate predictions. Binary cross-entropy is used to measure the loss function.

The mathematical equation for binary cross-entropy is as follows [30]:

$$L(y,\hat{y}) = -\left[y\log(\hat{y}) + (1-y)\log(1-\hat{y})\right] \tag{10}$$

where:

y is the actual label (either 0 or 1)

$\hat{y}$ is the predicted probability to the class labelled 1

The first term in the equation penalises the model if it predicts a low probability for a positive example, and the second term penalises the model if it predicts a high probability for a negative example. The overall loss is the sum of these two terms.

2.4 Dataset description

The ND-IIITD retouched faces dataset (Bharati et al., 2016) is obtained from the Notre Dame University, by providing signing biometric database release agreement. The dataset contains 4,875 face photos in total, 2,600 of which are original photographs and 2,275 of which have been altered shown in Table 1. The retouching is done using advanced software called Portrait Pro Studio Max. There are seven different sets of probe photos, each with unique portraits and examples of retouching, as shown in Figure 4.

Table 1 Dataset description

Dataset	*Real*	*Retouched*	*Total*
ND-IIITD	2,600	2,275	4,875

The level of modifications in the first two probe sets is lower than in other probes and they are only found in small areas. As the number of probes grows, the highest departure from the original images is shown with the seventh probe. There are 325 facial images total in each probe set, 211 of which are male and 114 of which are female. Equal ratio of real and retouched images is maintained to form the training and testing dataset. From the available dataset, the images are randomly selected for training, validation and test dataset. As per the batch size considered in the Table 2, the train, validation and test dataset is divided into 63, 34 and 06 batches respectively.

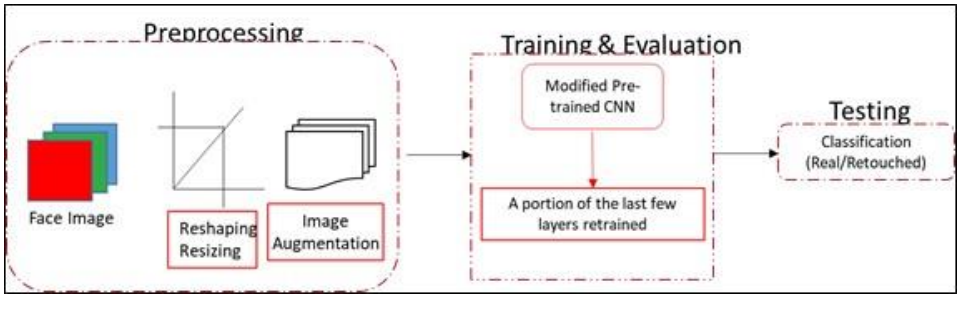
Figure 4 Original (real) and retouched images from each probe (see online version for colours)

Notes: (i) 1st row contains real images from preset 1 to 7 and (ii) 2nd row contains fake images from preset 1 to 7 respectively.

3 Proposed algorithm

The steps to classify the real and retouched images for the proposed model are as follows and shown in Figure 5.

Figure 5 Proposed model for detecting retouched face images



1 Prepare your dataset: This work includes dataset of images that are labelled with their corresponding classes. The dataset used over here is balanced means equal ratio of real and retouched probes is considered. The dataset is split into training, validation, and test sets. Hence, data augmentation is utilised by applying horizontal flip and 0.20-degree rotation to artificially yet realistic images to overcome the problem of overfitting.

2 Pre-process your images: VGG16 accepts images of size 224 × 224. Hence, data transformation is done over the training and validation dataset to rescale and fit the images as per VGG16 model. Transformation involves resizing images to a standard size and divided into batch of 32, converting them to grayscale or RGB, and normalising their pixel values.

3 Load the base model: VGG16 pre-trained model is load using a deep learning framework such as Keras or TensorFlow. The convolution layers of the model work on generalised features and top layers of the model works for task specific features.

4 Make new TL model: Hence, the changes are made only on top layers of VGG16. The FC layers of VGG16 are removed. A new FC layer is build up by adding Global average pooling, a dropout and a dense layer.

5 Compile and fit: The new model is trained using training set and evaluate its performance on the validation set. The convolution layers of pre-trained model are freeze during first training and only custom added FC layers will learn the feature maps from the given dataset.

6 Optimiser used during first training: During evaluation and first training either Adam or RMSprop optimiser is used to boost up the performance of the proposed model.

7 Fine-tune the model: Since the VGG16 model was pre-trained on a large dataset, it already has learned many features that can be used for image classification. However, we fine-tune the model by unfreezing some top convolution layers and re-train the new model. Hence, the weight of custom FC layers and unfreeze layers are updated during fine-tuning, which improves the performance of the model.

8 Optimiser used during fine-tune: During evaluation and fine tuning either Adam or RMSprop optimiser is used to boost up the performance of the proposed model.

9 Test the model: The accuracy of the model is tested on the test dataset. the class of the images of test dataset are predicted and normalised the values of prediction near to one of the values 0 or 1 using sigmoid function. As it is binary classification, threshold is defined to determine the predicted values as either 0 or 1.

3.1 Implementation details

Total four experiments are performed to find the accuracy and performance of the proposed model. The experiments are set based on the optimiser used during first training and fine-tuning of the TL pre-trained VGG16. The batch size, epoch, loss function and optimisers used during first training and fine-tuning of the model is considered as shown in Table 2 for the entire model evaluation. The optimisers used in this paper are Adam and RMSprop. Based on the optimisers used, the model is labelled as shown in the sub sequent Section 3.1.1 to 3.1.4. The training and validation accuracy and cross entropy loss are measure and compared for every listed (Table 3) combinations of optimiser. The motive behind using different pair of optimiser for classification, is the updating of the weights and biases is different for different optimisers which may lead the accuracy of the model to different measures.

3.1.1 Optimiser_11

As per Table 3, the optimiser_11 stands for the Adam optimiser used during first training of proposed model and the same optimiser is used during fine-tuning of the proposed model. The training and validation dataset is derived from ND-IIITD dataset. The hyper parameters are selected as shown in Table 2.

Table 2 Hyper parameters set for the proposed VGG16 TL model

Training mode	*Parameters*	*Parameters value*
First training	Optimiser	Adam/RMSprop
	Batch size	32
	Epoch	10
	Learning rate (LR)	0.001
	Criteria	Cross entropy loss
Fine-tune	Optimiser	Adam/RMSprop
	Batch size	32
	Epoch	20
	Learning rate (LR)	0.0001
	Criteria	Cross entropy loss

3.1.2 Optimiser_12

As per Table 3, the optimiser_12 stands for the Adam optimiser used during first training of proposed model and the RMSprop optimiser is used during fine-tuning of the proposed model. Hence, weight updating will be different during first training and fine-tune phase, which affect the accuracy, loss and performance parameters of the VGG16 TL Model. Hyper parameters selected for this pair of optimisers are same as Section 4.1.1.

Table 3 The optimiser pair used for classification of retouched face images

Proposed VGG16 TL model	*Optimiser to be used*	
	During 1st Training of model	*During Fine tune of the model*
Oprimiser-11	Adam	Adam
Oprimiser-12	Adam	RMSprop
Oprimiser-21	RMSprop	Adam
Oprimiser-22	RMSprop	RMSprop

3.1.3 Optimiser_21

As per Table 2, the optimiser_21 stands for the RMSprop optimiser used during 1st training of proposed model and Adam optimiser is used during fine-tuning of the proposed model. Hyper parameters selected for this pair of optimisers are same as Section 4.1.1.

3.1.4 Optimiser_22

As per Table 2, the optimiser_22 stands for the RMSprop optimiser used during 1st training of proposed model and the same optimiser is used during fine-tuning of the proposed model. Hyper parameters selected for this pair of optimisers are same as Section 4.1.1.

3.2 Training and evaluation of the VGG16 TL model

Tasks involving training the model and evaluation on the dataset are carried out on Google Colab using GPU runtime. We use tensor flow, an open source library to the data with 32 samples per batch, to preprocess the dataset and to load the pre-trained model VGG16 with 'ImageNet' weight. The proposed VGG16 TL is trained on train and validation dataset during initial training and fine-tuning. The fine tuning is started from epoch 10.

Figure 5 The training and validation accuracy and loss plot for proposed VGG16 TL model (fine tuning started from epoch 10) (see online version for colours)

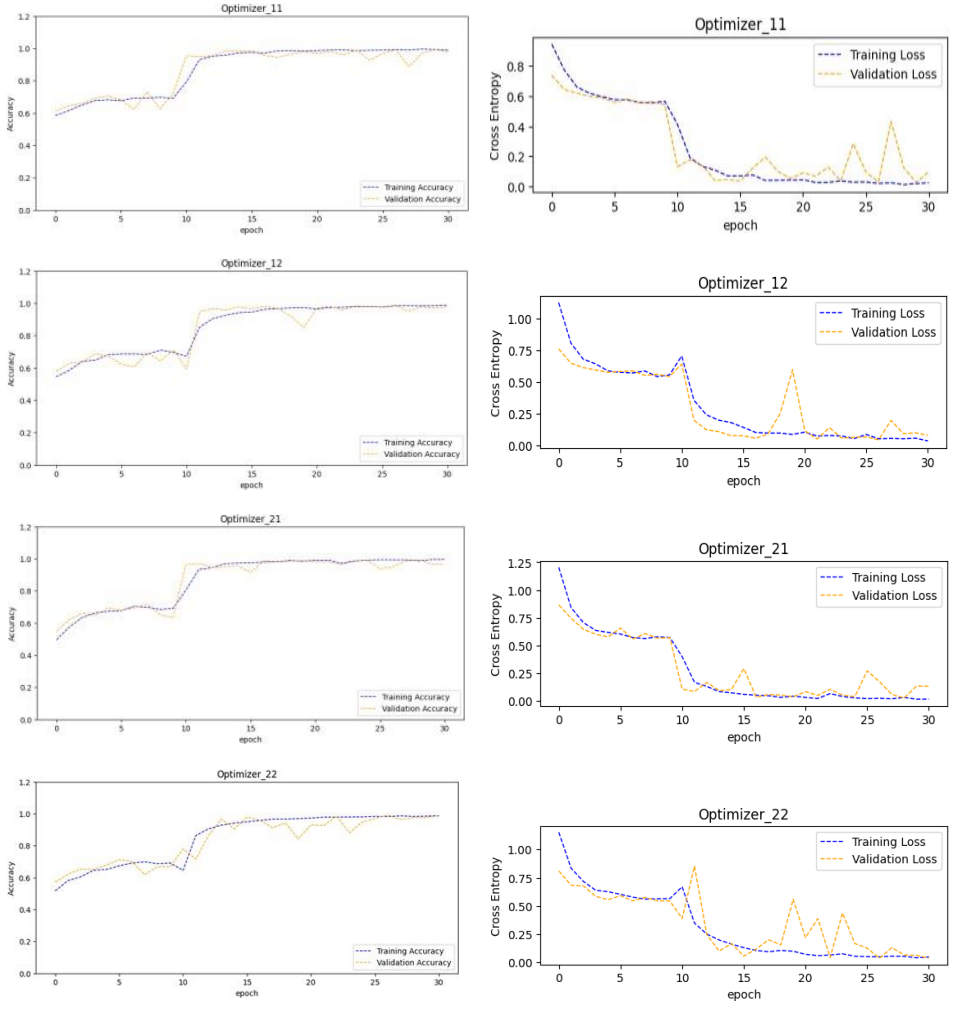


Figure 6 shows the comparative analysis of training and validation accuracy and loss over all 4 proposed models. From the figure it can be seen that the proposed model performs nearly equal on training accuracy for all 4 optimisers set over epoch 30. Notably, Optimiser_21 achieved the highest accuracy at epoch 30, reaching a perfect 99.54%. This represents a significant improvement compared to its accuracy of 49.54% at epoch 1, showcasing a remarkable 50% increment. But, the validation accuracy of the model is higher at epoch 30 for Optimiser_22 compared to others. It has been clearly seen that the cross entropy losses are keep on decreasing as epoch are increasing from 1 to 30 for training dataset. but, for validation dataset that decrement is not as smooth as the training dataset, and the minimum value of loss is achieved for Optimiser_22. The value of accuracy and loss over epoch 1 and Epoch 30 are also depicted in Table 4 and 5. The maximum increment of 50% is achieved in terms of training accuracy for Optimiser_21, whereas, validation accuracy is increased by 41.86% for Optimiser_22, as per Table 5.

Figure 6 Comparison of train and validation accuracy (1st column 2 figures) and loss (1st column bottom 2 figures) and confusion matrix (2nd column) for all optimiser sets of proposed VGG16 TL model (see online version for colours)

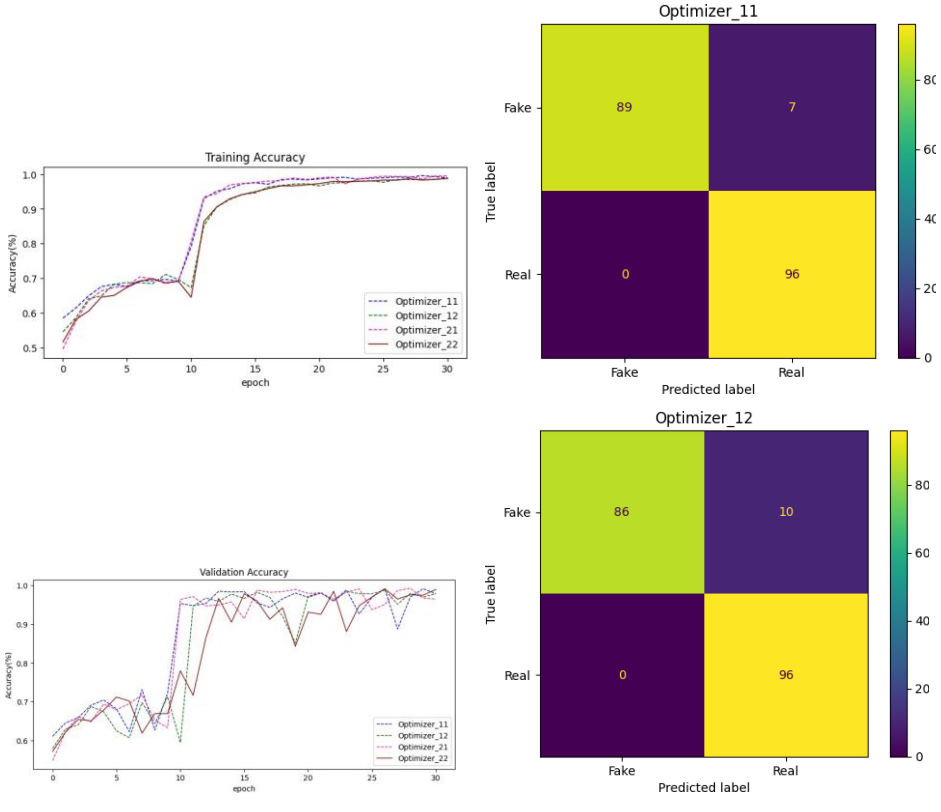


Figure 6 Comparison of train and validation accuracy (1st column 2 figures) and loss (1st column bottom 2 figures) and confusion matrix (2nd column) for all optimiser sets of proposed VGG16 TL model (continued) (see online version for colours)

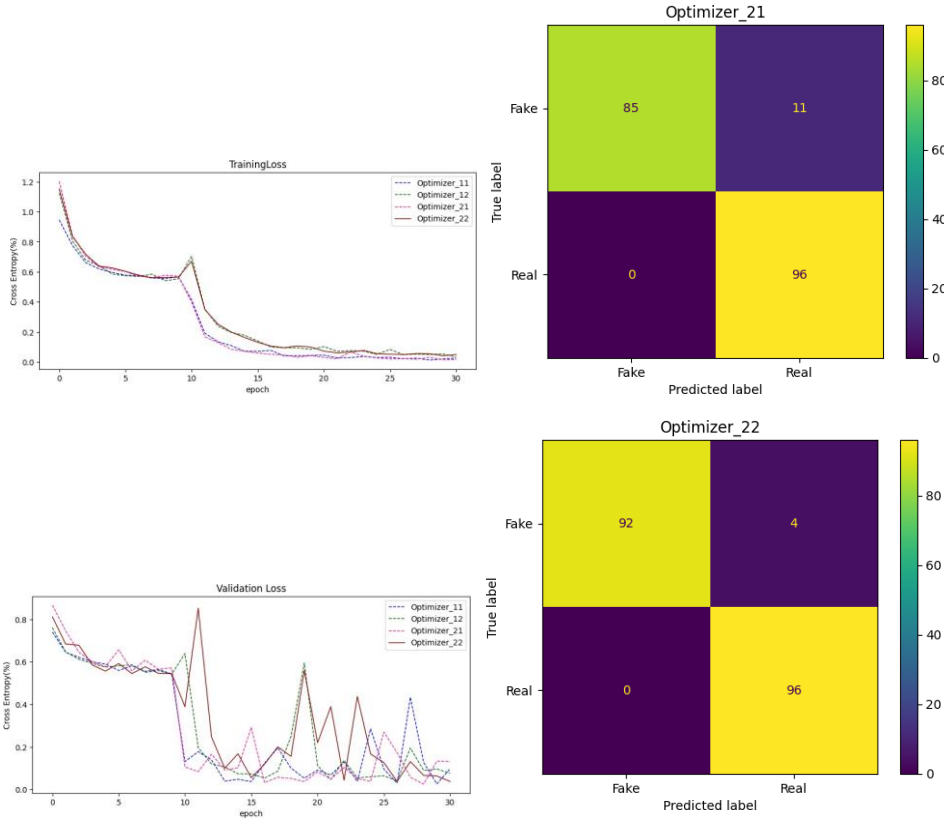


Table 4 Comparison of train accuracy (%) for different models

Proposed VGG16 TL model with optimiser set	*Accuracy @Epoch 1*	*Accuracy @ Epoch 30*	*Accuracy increment (%)*
Optimiser_11	0.5849	0.9896	40.47
Optimiser_12	0.5452	0.9893	44.41
Optimiser_21	0.4954	0.9954	50
Optimiser_22	0.5159	0.9878	47.19

Table 5 Comparison of validation accuracy (%) for different models

Proposed VGG16 TL model with optimiser set	*Accuracy @ Epoch 1*	*Accuracy @ Epoch 30*	*Accuracy increment (%)*
Optimiser_11	0.6102	0.9786	36.84
Optimiser_12	0.5786	0.9879	40.93
Optimiser_21	0.5479	0.9647	41.68
Optimiser_22	0.5712	0.9898	41.86

4 Result analysis and discussion

Several metrics have been devised to evaluate the effectiveness of the trained CNN (Uddin and Campus, 2021). A confusion matrix is built for classification tasks to evaluate the model quality; it classifies the predictions by model, based on whether they are correctly labelled the image class or not. Its four core principles are TP, TN, FP and FN called as a full true positive, true negative, false positive and false negative respectively. True is related with the label 1 and false concerned with label 0.

Based on the confusion matrix, the precision, recall, F1 score and accuracy parameters of the proposed VGG16 models are measured for the two classes (real and retouched) for classification.

Precision: number of correctly categorised/classified real/retouched faces among all the identified real/retouched cases.

$$P = TP / (TP + FP) \tag{11}$$

Recall is the number of correctly categorised/classified real cases from all the positive representations.

$$R = TP / (TP + FN) \tag{12}$$

F1 score is the harmonic average of precision and recall.

$$F1 = 2 * \left[(P * R) / (P + R)\right] \tag{13}$$

Accuracy, consequently, is determined as the ratio of correctly identified predictions over the total predictions.

$$A_{CC} = \frac{Correct predictions}{Total predictions} \tag{14}$$

Table 6 Various performance parameters comparison for various VGG16 TL model

Proposed VGG16 TL model with optimiser set	*Class*	*Precision*	*Recall*	*F1-score*	*Support*	*Accuracy*
Optimiser_11	Real	0.9320	1.0000	0.9648	192	0.9635
	Retouched	1.0000	0.9271	0.9622		
Optimiser_12	Real	0.9057	1.000	0.9505	192	0.9479
	Retouched	1.000	0.8958	0.9451		
Optimiser_21	Real	0.8972	1.0000	0.9458	192	0.9427
	Retouched	1.0000	0.8854	0.9392		
Optimiser_22	Real	0.96	1.0000	0.9796	192	0.9792
	Retouched	1.0000	0.9583	0.9787		

Notes: The numbers samples utilised for evaluation are 96 each for real and retouched images

The confusion matrix for all evaluated all models is shown in Figure 6 (column 2), shows the lowest FP numbers are achieved in Optimiser_22, indicating a more accurate classification in this regard. Whereas the proposed VGG16 model with optimiser_21

exhibit maximum FP number 11, reflect the comparative poor classification. In Table 6, a range of standard evaluation scores, with the 'Support' column indicating the number of image samples utilised for testing. Optimiser_22 achieved the highest precision, recall, and F1-score for both the 'Real' and 'Retouched' classes, with F1-scores of 97.96% and 97.92%, respectively. The proposed model for all optimiser sets exhibited outstanding recall scores for 'real' images. Notably, highest recall value achieved in Optimiser_22 is 95.83% for 'fake' images, as per Table 6.

4.1 Comparison result

In recent years, there has been a scarcity of research in the area of face retouching classification, prompting researchers to seek more effective solutions. In this study, we conducted a comprehensive analysis and comparison of our proposed model with an existing work, as presented in Table 7. The reference work (Bharati et al., 2016) achieved a classification accuracy of 87.10% and 81.90% over a balanced train-test ratio. In contrast, our approach achieved an impressive overall classification accuracy of 97.92%, demonstrating its resilience and effectiveness even when trained on an imbalanced data distribution. Through the evaluation of four experiments, it is evident that all the proposed optimiser sets consistently outperform the state-of-the-art (SOTA) in terms of classifying retouched and real images. The model's skill in accurately identifying facial retouching is shown by this. The vital role of TL and fine-tuning in increasing the model's classification abilities is also shown by our work. Additionally, the model's robustness and general performance are further enhanced by the careful selection and application of optimisers throughout both the training and fine-tuning stages.

Table 7 Comparison of classification accuracies (%) proposed model with existing studies for ND-IIITD retouched faces dataset

Sr. no.	*Ref no.*	*Method*	*Accuracy*	*Real*	*Retouched*
1	Bharti et al. (2016)	Unsupervised DBM	81.90%	74.30%	90.90%
		Supervised DBM	87.10%	81.10%	93.90%
2	Proposed VGG16 TL model	Optimiser_11 (Adam, Adam)	96.35%	100.00%	92.71%
		Optimiser_12 (Adam, RMSprop)	94.79%	100.00%	89.58%
		Optimiser_21 (RMSprop, Adam)	94.27%	100.00%	88.54%
		Optimiser_22 (RMSprop, RMSprop)	97.92%	100.00%	95.83%

5 Conclusions

In conclusion, this paper has demonstrated the effectiveness of deep learning algorithms, particularly TL approaches, in the context of image classification. TL emerges as a valuable tool, especially when work with limited datasets or constrained computational resources. The following can be concluded based on the findings of this study.

1 TL, utilising the pre-trained VGG16 CNN architecture, can achieve an impressive classification accuracy of 97.92% with just 30 epochs, underscoring its efficiency and suitability for this task.

2 We successfully minimise the overfitting issue, ensuring the model's robust performance, by carefully choosing a training hyper parameters to 10 epochs with a learning rate of 0.001 for initial training and 20 epochs with a lowered learning rate of 0.0001.

3 RMSprop stands out as the most effective option among the tested optimisers, producing remarkable training and validation accuracies of 99.54% and 98.98%, respectively.

Future research in this area may include deeper hyper parameter optimisation to improve model performance. Additionally, investigating the adaptability of this approach to different image classification tasks and datasets could provide valuable insights.

Acknowledgements

I am thankful to the Notre Dame University for providing the ND-IIITD retouched faces datasets for our research work. I also express my sincere gratitude to my PhD supervisor Dr Vishal S Vora for his constant support and guidance in line to the successful completion of this research work.

References

57 Celebrities Before And After Photoshop Who Set Unrealistic Beauty Standards (2017) [online] https://www.boredpanda.com/before-after-photoshop-celebrities/?media_id=554499 (accessed 2 Spetmber 2020).

Abate, A.F., Nappi, M., Riccio, D. and Sabatino, G. (2007) '2D and 3D face recognition: a survey', *Pattern Recognit. Lett.*, Vol. 28, No. 14, pp.1885–1906, DOI: 10.1016/j.patrec.2006.12.018.

Bharati, A., Singh, R., Vatsa, M. and Bowyer, K.W. (2016) 'Detecting facial retouching using supervised deep learning', *IEEE Trans. Inf. Forensics Secur.*, September, Vol. 11, No. 9, pp.1903–1913, DOI: 10.1109/TIFS.2016.2561898.

Bharati, A., Vatsa, M., Singh, R., Bowyer, K.W. and Tong, X. (2017) 'Demography-based facial retouching detection using subclass supervised sparse autoencoder', arXiv.

Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S. and Choo, J. (2018) 'StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation', *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp.8789–8797, DOI: 10.1109/CVPR.2018.00916.

Cyriac, S., Raju, N. and Ramaswamy, (2021) 'Comparison of full training and transfer learning in deep learning for image classification', in *Data Science and Security, Proceedings of IDSCS 2021*, pp.58–67.

Dantcheva, A., Chen, C. and Ross, A. (2012) 'Can facial cosmetics affect the matching accuracy of face recognition systems?', *2012 IEEE 5th Int. Conf. Biometrics Theory, Appl. Syst. BTAS 2012*, September, pp.391–398, DOI: 10.1109/BTAS.2012.6374605.

Eggert, N. (2017) *Is she Photoshopped? In France, They Now Have To Tell You*, BBC News, September, pp.1–11, [online] https://www.bbc.co.uk/news/world-europe-41443027 (accessed 2 August 2020).

Gupta, A. (2021) *A Comprehensive Guide on Optimizers in Deep Learning*, Blogathon [online] https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers (accessed 15 October 2020).

Gupta, S. (2005) 'Digital alteration of photographs and intellectual property rights', *J. Intellect. Prop. Rights*, November, Vol. 10, pp.491–498, [online] https://www.manupatra.com (accessed 2 October 2020).

Hussain, M., Bird, J.J. and Faria, D.R. (2019) 'A study on CNN transfer learning for image classification', *Adv. Intell. Syst. Comput.*, June, Vol. 840, pp.191–202, DOI: 10.1007/978-3-319-97982-3_16.

Jain, A., Singh, R. and Vatsa, M. (2018) 'On detecting GANs and retouching based synthetic alterations', DOI: 10.1109/BTAS.2018.8698545.

Jain, A.K. (2014) Vol. 36368, pp.1–40, [online] https://wwwpapers3://publication/uuid/17FEF3DC-D6F9-408B-96E5-871ED133A4F1 (accessed 1 October 2020).

Kandel, I. and Castelli, M. (2020) 'Comparative study of first order optimizers for image classification using convolutional neural networks on histopathology images', *J. Imaging*, Vol. 6, No. 92, pp.1–17, DOI: https://doi.org/10.3390/jimaging6090092.

Kee, E. and Farid, H. (2011) 'A perceptual metric for photo retouching', *Proc. Natl. Acad. Sci. U. S. A.*, Vol. 108, No. 50, pp.19907–19912, DOI: 10.1073/pnas.1110747108.

Kose, N., Apvrille, L. and Dugelay, J.L. (2015) 'Facial makeup detection technique based on texture and shape analysis', *2015 11th IEEE Int. Conf. Work. Autom. Face Gesture Recognition, FG 2015*, DOI: 10.1109/FG.2015.7163104.

Antony, K. (2021) 'Fashion photographers and their rights', *Fash. Law J.*, pp.1–6.

Li, H., Siddiqui, O., Zhang, H. et al. (2019) 'Joint learning improves protein abundance prediction in cancers', *BMC Biol.*, Vol. 17, p.107, https://doi.org/10.1186/s12915-019-0730-9.

Parkhi, O.M., Vedaldi, A. and Zisserman, A. (2015) *Deep Face Recognition.pdf*, No. Section 3.

Ramdan, A., Heryana, A., Arisal, A., Kusumo, R. and Pardede, H. (2020) 'Transfer learning and fine-tuning for deep learning-based tea diseases detection on small datasets', pp.206–211, DOI: 10.1109/ICRAMET51080.2020.9298575.

Rathgeb, C. et al. (2020) 'PRNU-based detection of facial retouching ISSN 2047-4938', *IET Biometrics*, July, Vol. 9, No. 4, pp.154–164, DOI: 10.1049/iet-bmt.2019.0196.

Rathgeb, C., Dantcheva, A. and Busch, C. (2019) 'Impact and detection of facial beautification in face recognition: an overview', *IEEE Access*, Vol. 1, DOI: 10.1109/ACCESS.2019.2948526.

Sharma, K., Singh, G. and Goyal, P. (2022) 'IPDCN2: improvised patch-based deep CNN for facial retouching detection', *Expert Systems with Applications*, Vol. 211, p.118612, DOI: 10.1016/j.eswa.2022.118612.

Shyu, M., Chen, S. and Iyengar, S.S. (2018) 'A survey on deep learning : algorithms, techniques', *ACM Comput. Surv.*, Vol. 51, No. 5, pp.1–36.

Simonyan, K. and Zisserman, A. (2015) 'Very deep convolutional networks for large-scale image recognition', *3rd Int. Conf. Learn. Represent, ICLR 2015 - Conf. Track Proc.*, pp.1–14.

Singh, A., Tiwari, S. and Singh, S.K. (2013) 'Face tampering detection from single face image using gradient method', *Int. J. Secur. its Appl.*, Vol. 7, No. 1, pp.17–30.

Singh, R., Vatsa, M., Bhatt, H.S., Bharadwaj, S., Noore, A. and Nooreyezdan, S.S. (2010) 'Plastic surgery: a new dimension to face recognition', *IEEE Trans. Inf. Forensics Secur.*, Vol. 5, No. 3, pp.441–448, DOI: 10.1109/TIFS.2010.2054083.

Smith, J. (2022) *Photoshop Law requires Retouching Disclosure*, American Graphic Institute [online] https://www.agitraining.com/adobe/photoshop/classes/photoshop-law-requires-retouching-disclosure (accessed 1 December 2022).

Stoumpou, V., Vargas, C.D.M., Schade, P.F., Boyd, J.L., Giannakopoulos, T. and Jarvis, E.D. (2023) 'Analysis of Mouse Vocal Communication (AMVOC): a deep, unsupervised method for rapid detection, analysis and classification of ultrasonic vocalisations', *Bioacoustics*, Vol. 32, No. 2, pp.199–229, DOI: https://doi.org/10.1080/09524622.2022.2099973.

Uddin, J. and Campus, S. (2021) 'Fish survival prediction in an aquatic environment using random forest model', September, Vol. 2021, pp.614–622, DOI: 10.11591/ijai.v10.i3.pp614-622.