Multimedia Tools and Applications (2023) 82:29943–30001
https://doi.org/10.1007/s11042-022-14165-4

TRACK 2: MEDICAL APPLICATIONS OF MULTIMEDIA

Detection of Diabetic Retinopathy using Convolutional Neural Networks for Feature Extraction and Classification (DRFEC)

Dolly Das [1] · Saroj Kumar Biswas [1] · Sivaji Bandyopadhyay [1]

Received: 17 March 2022 / Revised: 14 June 2022 / Accepted: 27 October 2022 /
Published online: 29 November 2022

Abstract
Diabetic Retinopathy (DR) is caused as a result of Diabetes Mellitus which causes development of various retinal abrasions in the human retina. These lesions cause hindrance in vision and in severe cases, DR can lead to blindness. DR is observed amongst 80% of patients who have been diagnosed from prolonged diabetes for a period of 10–15 years. The manual process of periodic DR diagnosis and detection for necessary treatment, is time consuming and unreliable due to unavailability of resources and expert opinion. Therefore, computerized diagnostic systems which use Deep Learning (DL) Convolutional Neural Network (CNN) architectures, are proposed to learn DR patterns from fundus images and identify the severity of the disease. This paper proposes a comprehensive model using 26 state-of-the-art DL networks to assess and evaluate their performance, and which contribute for deep feature extraction and image classification of DR fundus images. In the proposed model, ResNet50 has shown highest overfitting in comparison to Inception V3, which has shown lowest overfitting when trained using the Kaggle's EyePACS fundus image dataset. EfficientNetB4 is the most optimal, efficient and reliable DL algorithm in detection of DR, followed by InceptionResNetV2, NasNetLarge and DenseNet169. EfficientNetB4 has achieved a training accuracy of 99.37% and the highest validation accuracy of 79.11%. DenseNet201 has achieved the highest training accuracy of 99.58% and a validation accuracy of 76.80% which is less than the top-4 best performing models.

✉ Dolly Das
das.dolly0019@gmail.com

Saroj Kumar Biswas
bissarojkum@yahoo.com

Sivaji Bandyopadhyay
sivaji.cse.ju@gmail.com

[1] Department of Computer Science and Engineering, National Institute of Technology Silchar, Cachar, Silchar, Assam 788010, India

Keywords Diabetic Retinopathy · Fundus image · Convolutional Neural Network · Deep Learning · Image classification

1 Introduction

Diabetic Retinopathy (DR) is an eye disorder which is a consequence of Diabetes Mellitus (DM). DR causes inflammation and breach of retinal blood vessels for the formation of various irregular retinal lesions. It is frequently observed in patients having diabetes from a longer duration of time such as 10–15 years [47]. Statistics have stated that 80% of diabetic patients, suffering from protracted diabetes suffer from diverse phases of DR [28]. India has approximately 73 million individuals suffering from diabetes [77]. The early consultation of the patient with diabetes, with experts for DR assessment and evaluation has become necessary. The process of early diagnosis can reduce the development of DR and susceptibility to severe blindness [22]. If the disease remains undiagnosed, DR progresses in forms of various stages through formation of lesions such as Microaneurysms (MAs), Exudates (EXs), Hemorrhages (HEs), Cotton Wool Spots (CWSs), abnormal structure of the Optic Disc (OD), abnormal Foveal Avascular Zone (FAZ), Neovascularizations, Intra Retinal Microvascular Abnormalities (IRMAs), and many more [24–26, 52, 73, 76, 84]. The presence of these lesions can lead to vision problem and as the disease progresses to severe stages, it can lead to complete blindness [22, 73].

The experts of the domain, called as Ophthalmologist examines the human retina using a high-resolution digitized fundus camera for capturing fundus image. The fundus perceives various DR retinal lesions for image annotation and follow-up necessary treatments. The process of physical diagnosis is painstaking, and the dearth of unavailability of proper means for treatment makes early detection of DR, a challenging task. Thus, such a medical disorder necessitates advanced means for proper diagnosis and treatment. To overcome the challenges of labor-intensive DR detection, researchers have proposed intelligent expert systems, using Deep Learning (DL) for analysis and in-depth study of DR features from fundus images. Intelligent systems [4] are effective with respect to time, feature extraction, error recognition, and early diagnosis and treatment compared to traditional methods. These smart systems take fundus images as input, which are enhanced and analyzed for extraction of significant features, for classification of DR as moderate DR, moderate NPDR, mild DR, severe NPDR, mild Non-Proliferative DR (NPDR), early Proliferative DR (PDR), PDR, etc. [29].

Thus, intelligent systems using DL are proposed as an early and potentially scalable alternative for DR detection. Conventional ML models and data analysis approaches are shallow in nature, and have shown poor performance in learning and training complex non-linear features from larger datasets and hence, are unable to exhibit better analysis and interpretation [41]. Besides, DL based CNN models [7] have overshadowed ML models in performance, through inbuilt preprocessing, convolutional operations, better learning and generalization with deeper networks [88], less overfitting [88], data imbalance mitigation [88], optimization etc. Therefore, various state-of-the-art ML-based models [12, 49, 50] called DL models are proposed for deep feature extraction and image [6, 7] classification tasks. The inherent caliber of such models due to their hierarchical structures, enhances the learning of the model, to achieve state-of-the-art performance. In diagnosis of DR, different DL architectures proposed earlier, have exhibited different feature extraction and classification performances. However, in a single experimental set up, the effectiveness of assessing a comprehensive DL

model is not exhaustive, and is unreliable. Therefore, this paper proposes a DL model entitled Diabetic Retinopathy Feature Extraction and Classification (DRFEC), an intelligent expert system using DL architectures, for an exhaustive comprehensive evaluation for detection of DR. The proposed approach is an intensive meta-analysis of 26 pretrained DL networks for identification of the optimal architecture suitable for a larger dataset such as Kaggle EyePACS DR detection with significant skewness, and thereby identify architectural patterns and corresponding loopholes in performance, to ease the process of traditional methods of detection. The model identifies suitable architecture-optimized goals to mitigate overfitting and poor generalization. The main contributions of the manuscript have been enlisted below:

1. The proposed model performs a comprehensive comparative assessment and evaluation to determine the behaviour of 26 DL models on the DR dataset
2. To conduct meta-analysis of traditional as well as state-of-the-art DL architectures on the highly skewed Kaggle DR detection dataset, with minimal resources and identify the best amongst them for future application and analysis.
3. The proposed model identifies high bias and high variance during DR image classification to identify the best DL classifier

This manuscript is sectionalized into various sections. Section 2 is an analysis of various related works and models proposed earlier, for identification of loopholes in early DR detection. Section 3 gives a detailed illustration of the proposed methodology, and establishes the significance of DL models for better feature extraction and classification. Section 4 illustrates the significance of the 26 DL models, based on the analysis and comparison of their performances, for determination of an optimal DL architecture(s) for DR detection. Section 5 concludes on a note identifying the suitable DL architecture(s) for DR detection.

2 Literature review

DR is a chronic health disease which requires early detection and treatment [48]. It is important to identify DR using an intelligent system for faster prediction since manual examination and detection of the disease are unreliable and highly prone to error. Therefore, various researchers and medical experts have adopted and approached for advanced feature extraction and image classification, for early DR detection. Thus, various works have been proposed in this respect using ML and DL techniques, and in which DL techniques have completely outperformed ML based models on grounds of processing large dataset, efficient computation, overfitting, generalization and better prediction. This section introduces some of the recognized works on image domain using DL techniques especially upon fundus images for early DR detection [32].

Wang et al. [85] have proposed a boosted CNN architecture using EfficientNet B3 for extraction of images features using integrated attention and feature fusion-based mechanisms, random center cropping upon Rectified Patch Camelyon (RPCam) datasets to predict and classify lymph node metastasis in breast cancer images. The model is compared with baseline models such as EfficientNet B3, ResNet50 and DenseNet121. Gurcan et al. [31] have proposed an automated DR classification system based on preprocessing, feature extraction, and classification steps using deep CNN and ML methods. The model has extracted features from a pre-trained InceptionV3 model using transfer learning. The model has compared various ML methods namely Bagged Decision Trees, XGBoost, Random Forest, Extra Trees,

Support Vector Machines, Logistic Regression, and multilayer perceptron in which XGBoost performs better. The model has used Grid search and calibration for analysis in addition to comprehensive preprocessing and fine-tuning. The model has extracted generic descriptors from one of the initial layers of InceptionV3, without layer-wise tuning and has achieved competitive classification accuracy with ML methods. It is observed that the descriptors are obtained and extracted from the initial layers of CNN which have comparatively low-level information than the higher layers of a CNN. This may be inefficient in the process of DR detection and hinder the detection of actual features thereby affecting the robustness of the model. Sarki et al. [69] have proposed a methodical study which identifies and signifies various preprocessing operations such as Contrast-Limited Adaptive Histogram Equalization (CLAHE), morphological operations, image segmentation for blood vessel segmentation etc. for Diabetic Eye Disease (DED) detection. The model has used a mini-batch size of 32, cross-entropy loss function, Adam and RMSprop upon the datasets namely DRISHTI-GS, Messidor, Retinal Dataset and Messidor-2. The proposed automated classification framework has used image enhancement, image augmentation and segmentation, and classification using very small datasets having poor category DR samples. Mayyaa et al. [56] have proposed a methodical review to examine the diagnostic use of automated microaneurysm detection for DR, and has identified various strengths and weaknesses. The methodologies appraised in this article confronts challenges that needs to be addressed in designing an effective algorithm for early diagnosis. Hattiya et al. [33] have appraised AlexNet DL mechanism as an ideal CNN architecture for DR detection and compared diverse CNN architectures namely MobileNet, DenseNet201, InceptionV3, ResNet50, NASNetMobile and MNASNet, using 23,513 retina images.

Kamal et al. [44] have anticipated a transfer learning model for DR detection which mitigates imbalanced dataset. The model has fine-tuned Inception V3, VGG19, ResNet50, MobileNet, ResNet50V2, DenseNet121, MobileNetV2 and NASNetMobile using COVID-19 [5, 8] and Pneumonia datasets. Islama et al. [40] have reported a comprehensive systematic review of the performance of DL algorithms gauged precisely for computerized DR detection in fundus images. Lee et al. [53] have proposed a transfer learning NASNet-A (large) to extract bottleneck features from spectral-domain optical coherence tomography images and an ensemble training for prediction. Bodapati et al. [10] have anticipated a DR model using transfer learning, and feature extraction using VGG-16, Inception ResNetV2, Xception and NASNet to boost feature exemplification which are compared with handcrafted features, for DR detection. The model has compared feature fusion and pooling approaches and have identified averaging pooling simple fusion approach upon Deep Neural Networks (DNN) as effective in performance. It has extracted features of DR images collected from Kaggle APTOS 2019 contest dataset, using VGG16 and Xception. The authors blended these features to get the final feature representations, which are used to train DNN. Shah et al. [71] have proposed a DCNN model to distinguish referable DR using 1533 macula centered fundus images and MESSIDOR dataset. The model compares the evaluation process of ground truth data and machine learned data. The algorithm is a composition of three-version based training modules using 80,000, 96,500, and 112,489 unidentified fundus images, respectively. Pour et al. [62] have proposed EfficientNet based feature extraction and classification model using EfficientNet B5 for DR detection using MESSIDOR, MESSIDOR-2 and IDRiD datasets. The images are preprocessed using CLAHE, and the model has achieved an AUC of 0.945 on MESSIDOR, and AUC 0.932 on IDRiD.

Chetoui et al. [14] have proposed an EfficientNet based feature extraction and classification model using EfficientNet B7 and Global Average Pooling, for DR detection. The model has used Kaggle EyePACs and APTOS 2019 datasets, and have extracted features such as EXs, HEs and MAs using Gradient-weighted Class Activation Mapping (Grad-CAM). Tymchenko et al. [83] have proposed a DCNN encoder-based feature extraction for DR detection using pre-trained EfficientNet-B5, EfficientNet-B4, SE-ResNeXt50 and ensemble of 20 models. Chaturvedi et al. [13] have proposed a modified DenseNet121 network on APTOS 2019 dataset and uses 3662 fundus images, for DR detection. Samanta et al. [66] have anticipated a fine-tuned DenseNet121 model which uses 3050 images for training, and is tested using VGG16, InceptionV1, InceptionV3, InceptionV2, Xception, AlexNet, ResNet-50 and DenseNet for DR detection. Sarki et al. [68] have proposed an inclusive assessment of 13 pretrained CNNs, using MESSIDOR and Kaggle dataset, for detection of DR. Ji et al. [42] have proposed an augmented DNN model using Inception V3, DenseNet121 and ResNet50 for transfer learning and improve computational proficiency and classification, for DR detection. The model has proposed various subnetworks for each of the DNN and analyzed their performance on large OCT image datasets of 83,484 images, for detection of DR lesions. The proposed model achieves accuracy and stability for InceptionV3 and ResNet50, however in case of DenseNet121, no significant improvement could be seen due to its dense connection architecture.

Michele et al. [57] have proposed a fine-tuned pretrained feature extraction and classification model using MobileNetV2, dropout and a linear SVM classifier, for palmprint recognition. The model has used a PolyU palmprint dataset of 6000 images. Hui et al. [38] have anticipated a modified extreme inception-based U-Net segmentation module, to extract effective features using multitask learning and distance representation, from remote sensing images. The model has accomplished better performance with multitasking, on the datasets. Jiang et al. [43] have proposed an interpretable ensemble ResNets based feature mining and classification for DR detection using a DL model which constitutes Inception V3, ResNet152 and InceptionResNetV2, and Adaboost algorithm. The model has employed 28,244 training images, and achieved better performance using the integrated DL model than the individual models. Orlando et al. [59] have proposed an ensemble LeNet-CNN approach for the detection of MAs, HEs, red lesions for DR detection, using hand-crafted features, CNN features and a combination of both. The proposed model has achieved better results with the combination of CNN and handcrafted features. Suriyal et al. [78] have proposed a real-time DR model using MobileNet, on internet-deprived portable devices, for DR detection. The model has used 16,798 images.

Huang et al. [37] have proposed a compact novel network architecture called CondenseNet which combines dense connectivity with novel learned group convolutions, using CIFAR-10, ImageNet and CIFAR-100 datasets, for image classification. The novel CNN architecture has achieved cost-efficient performance in comparison to MobileNets, DenseNet-190 and ShuffleNets. Pogorelov et al. [61] have compared global features extracted from gastrointestinal tract images using transfer learning models such as ResNet50 and Inception V3, and proposed a modified CNN. The model has compared the predictions of ML and DL classifiers, and has achieved better performance using ResNet50 than Inception V3 for feature extraction. Gulshan et al. [30] have proposed a DL InceptionV3 for detection of DR and diabetic macular edema in fundus images. The proposed methodology has deployed the EyePACS-1 dataset and MESSIDOR-2 dataset. The model has used an ensemble of ten networks and their linear average is used for prediction, upon 128,175 images. Nneji et al. [58] have proposed a two-channel

preprocessing weighted fusion deep learning network on fundus images which uses CLAHE fundus images and the contrast-enhanced canny edge detection (CECED) fundus images, from 2000 Kaggle images and MESSIDOR dataset for the detection of DR. The model uses VGG-16 and a modified Inception-V3 for feature extraction and merges the channel output using a weighted fusion approach. The model performs a comprehensive analysis on six different baseline models, reduces the kernel of the Inception module B to (4 × 4), and improperly manipulates the size of the layers and the symmetricity of the image. Dong et al. [23] have proposed a DL model using InceptionV3 and VGG-16 for the detection of DR which is compared with mainstream models such as GoogLeNet, AlexNet and ResNet50 using 2693 wide-field optical coherence tomography augmented images. The real-time dataset employed is expensive and time-consuming. The authors have claimed that wide-field optical coherence tomography images are better than optical coherence tomography images due to incorporation of a better field-of-view (FOV). The model is built upon conventional DL models with a very limited dataset, which is unreliable.

Padmanayana and Anoop [60] have proposed a CNN model for the detection of DR through comparison of the performance of various optimizers such as Adagrad, RMSProp with momentum and Adam, using the APTOS 2019 Kaggle dataset for training and 1000 images collected from a private institute for testing. It uses weighted CLAHE, Gaussian blur and Ben Grahams fraction maxpooling for preprocessing. Sivapriya et al. [75] have proposed a Recurrent Neural Network (RNN) to identify hard EXs for the detection of DR. The model uses limited data of 400 images from the MESSIDOR dataset and performs various preprocessing operations on them without any significant changes in the architecture of the RNN. It has compared the performance of the approach with some of the earlier works and reflects a conventional and unreliable behaviour. Bora et al. [11] have proposed an automated risk analysis system for the detection of development of DR using fundus images, in patients having diabetes with no DR. It reports a risk stratification tool and learns the various associated risk factors causing the development of DR from the stage of no DR. It uses a large retrospective longitudinal dataset for the analysis- 575,431 eyes/single images of the development set has 28,899 known outcomes, 546,532 to augment the training process via multitask learning, 3678 images in the internal validation set and 2345 images in the external validation set. Deepa et al. [18] have proposed a multistage ensemble DCNN using InceptionV3 and Xception which concatenates multi-stage patch based and image-based probability vectors for deep feature extraction and SVM based ensemble classification, for the detection of DR. It uses voting and stacking for probability vector concatenation which is classified using an Artificial Neural Network (ANN). An ensemble of SVM classifier is used for further prediction on the ensembled classification output. It is observed that the proposed architecture incorporates a single-tier ensemble for feature extraction and a double-tier ensemble using different classifiers for classification which makes the overall architecture computationally expensive and time consuming.

Saeed et al. [65] have developed a two-stage Principal Component Analysis based transfer learning algorithm and initializes the pretrained model with extracted (64 × 64) ROIs of MAs, EXs and normal features of the fundus images of Kaggle's EyePACs and MESSIDOR datasets, for detection of DR. It introduces a fixed-dimension based adaptive maxpooling to predict the label of the ROI, and compares the performance of DL models namely VGG-19, ResNet152 and Dual Path Network 107(DPN107) in which the fully connected layers are replaced with PCA for unsupervised feature discrimination. It uses Decision Tree (DT), RF and Gradient Boosting (GB) for classification and compares the overall performance of ResNet152 with all the other models. Tsai et al. [82] have proposed a DL model using

Inception V3, ResNet101 and DenseNet121 on global Kaggle's EyePACs dataset and local dataset from Taipei City Hospital (TCH), for DR detection and sets a higher overestimation rate on local dataset than global dataset due to differences in regional and ethnic factors. The proposed model uses a significant dataset for the DL models employed. However, it is biased and takes ethnic factors into account for the detection of DR and fails to learn region specific features. On comparison, DenseNet121 has performed better than InceptionV3 and ResNet101.Atwany et al. [3] have performed a literature review and analyses the significance of supervised, semi-supervised and vision transformer methodologies and learning paradigms and their significance in application to DL models for DR detection. It uses Kaggle's EyePACs dataset of 88,702 images, DDR dataset of 13 K images and 30,244 fundus images from Beijing Tongren Eye Centre for the study and have concluded that supervised learning is not a good paradigm for noisy data. Instead, the study identifies Semi-Supervised Learning (SSL) effective but unexplainable, and less prone to inductive bias to be able to handle variance due to cross domain shift. However, it identifies SSL as non-robust for small-scale datasets. The study also reviews various amalgamation techniques for synthesis of dataset such as Generative Adversarial Network (GAN) and Variational Autoencoders (VAE), and identifies less complex DL attention models such as Vision Transformers (ViT).

Das, S. et al. [16] have proposed a two-way CNN classifier based on Squeeze-and-Excitation memory module and CNN, using DIARETDB1 and a local dataset, for DR detection. The model introduces the significance of data augmentation in better model performance. Lim et al. [55] have performed a literature review on gradient-based interpretability methods in DL models such as saliency map, integrated gradient, layer-wise relevance propagation, occlusion testing, sensitivity analysis, class activation map, gradient-weighted class activation map and layer-wise relevance propagation, in the detection of DR. It identifies the drawbacks of these interpretability methods in detection of correct lesions for a given class and the lack of reliable ground truth. AbdelMaksoud et al. [1] have proposed an integrated CNN model called E-DenseNetBC-121 which is a combination of EyeNet [63] and DenseNet [39] and uses datasets such as EyePACS, IDRiD, MESSIDOR and APTOS 2019 for the detection of DR. Li et al. [54] have use a DL algorithm-based software for grading 1674 images for the detection of DR. However, it fails to detect DME which is responsible for DR, for early DR detection. It uses 1, 40,000 fundus images from publicly available EyePACs dataset and 1200 fundus images from Shanghai General Hospital. Deepa et al. [19] have proposed a comprehensive two-phase feature extraction algorithm which uses Xception and textural and transform based techniques to detect MAs for DR detection. It uses Siamese Network based CNN to perform hierarchical clustering along with Xception-based cluster selection, and a fine-tuned Xception model for extraction of patch-wise local and global features, respectively. The features are extracted from 2290 images and are classified using a Radial Basis Function (RBF) kernel based SVM which is compared with other classifiers such as RF, Adaboost and Multilayer Perceptron (MLP).

Sau and Bansal [70] have proposed a Fitness based Newly Updated Grasshopper Optimization Algorithm (FNU-GOA) for the optimization of a DL model and to optimize the threshold value in active contour method for the segmentation of blood vessels, MAs, EXs and HEs for DR detection. It is compared with several other optimization algorithms such as Particle Swarm Optimization (PSO), Grey wolf optimization algorithm (GWO), Whale optimization algorithm (WOA) and Grasshopper Optimization Algorithm (GOA), and ML classifiers such as Neural Network (NN), RNN, Long Short Term Memory (LSTM) and Deep NN. The meta-heuristic optimized algorithm achieves a poor specificity and fails to classify

negative samples correctly. Shaik and Cherukuri [72] have proposed a multi-stage end-to-end DNN pipeline called Hinge Attention Network (HA-Net) using gated attention VGG-16 discriminator and reconstruction autoencoder to produce attention maps, for learning latent representations in 3662 images of Kaggle's APTOS 2019 dataset, and ISBI-2018 IDRiD[2] dataset for the detection of DR using various optimizers. It achieves a poor accuracy on limited-graded dataset. The model employs various attention descriptors with minimum samples and suffers from the curse of dimensionality. It fails to generate a decision boundary and learn inter-spatial and inter-channel correlation present in latent features. It also fails to reduce latent spatial representations learned from baseline models such as VGG-16, VGG-19, ResNet50, ResNet50V2, Xception, MobileNet, Inception V3 and InceptionResNetV2, and from overlapping data.

On the basis of the literature review, it is clear that conventional models are outperformed by state-of-the-art DL models. Many conventional models are shallow in nature, in contrast to DL models where DL models have shown convincing results. Based on this inspiration, the proposed DRFEC aims to perform an exhaustive analysis on DR image classification to define the characteristics, behaviour and pattern of DR data, and to identify, interpret and implement a convincing model for the problem. The use of conventional modes with a very small and limited dataset is the main drawback in DR detection. In addition to this, the use of the training set of Kaggle's EyePACS dataset is very limited due to data imbalance and lack of features. Besides, processing a larger dataset through increase in the number of samples and data augmentation causes data explosion which is a huge challenge with constraints in adequate resources. Various research works have considered only a few images which ranges between a few thousand images. The existing models have shown resemblance in their performances through the use of a similar number of images with no significant improvement in the pattern of the collected data. Moreover, these models also cannot generalize real-time datasets because of biasness towards regional and ethnic factors, which remains an unsolved problem. They have also shown high variance thereby reflecting the poor learning process of the methodologies for the detection. The proposed model thus aims to train a DL model to gain insights on the problem, and identify and differentiate parameters responsible for various performance with respect to (w.r.t.) the optimal model. It aims to solve critical problems through identification and analysis of DR data from the inception using an imbalanced dataset and gradually optimizing the proposed baseline framework through hyperparameter tuning, model training and evaluation [64].

3 The proposed DRFEC

The manual examination of DR is a feasible but lethargic process, and hence not recommendable for early DR detection. Therefore, it is essential to examine DR using a proficient system which employs ML techniques such as DL for better detection. Various explorative works have been performed earlier for DR detection, using a variety of datasets and especially using the Kaggle's EyePACS dataset. The proposed DRFEC accomplishes inbuilt preprocessing, deep feature extraction and image classification using a comprehensive DL model which constitutes VGG-19, VGG-16 [74], Xception [15], InceptionV3 [79], MobileNet [35], MobileNetV2 [67], EfficientNet B0-B7 [81], DenseNet121, DenseNet169, DenseNet201 [36], ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2 [34], NASNetLarge [89], NASNetMobile [89] and InceptionResNetV2 [80]. The model uses

popularly known ImageNet [51] pre-trained DL models for training and evaluation of Kaggle DR Detection dataset obtained from EyePACS [17, 21].

DL models have the proficiency to extract new and deep features, from previously learned features using representation learning. Thus, the proposed model emphases on finding scope for various improvements upon the data and the model, and thereafter determines the most suitable network amongst them for DR image classification. Figure 1 depicts a pictorial demonstration of DRFEC deploying DL models. It is implemented using a 64-bit operating system, ×64-based processor, Windows 10 Pro, DL package Keras from TensorFlow, Python 3.8, TensorFlow version 2.4, 64GB RAM and Intel(R) Xeon(R) W-2155 CPU @ 3.30GHz 3.31 GHz. The outline of the different steps of DRFEC is given in Fig. 1 below.

The proposed DRFEC processes the input data acquired from Kaggle repository and downsamples it accordingly based on different architectures employed in the model. Then the employed CNN is used to perform feature extraction and multi-class classification of DR fundus images. Flowchart 1 is an illustration of the working principle of DRFEC where each time a CNN is chosen to train the model. Every CNN employed is individually used to compute the training and validation/test accuracy values to check for model overfitting and generalization. Based on the performances and inferences drawn from every single DL CNN model, the best DL architecture for deep DR feature extraction and image classification is extracted.

3.1 Experimental environment

The proposed framework is implemented using a 64-bit operating system, x64-based processor, Windows 10 Pro, DL package Keras from TensorFlow, Python 3.8, TensorFlow version 2.4, 64GB RAM and Intel(R) Xeon(R) W-2155 CPU @ 3.30GHz 3.31 GHz.

3.2 Image dataset acquisition

In DRFEC, the model is assessed using a large dataset of fundus images, to analyze the performances of 26 DL architectures. The model uses Kaggle's EyePACS dataset and is exceedingly disproportionate [20]. The model uses 35,126 images, and is trained using 27,446

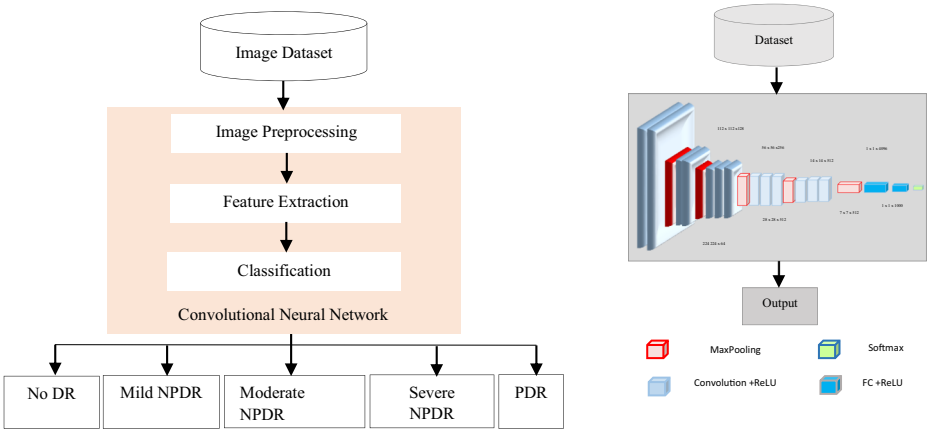
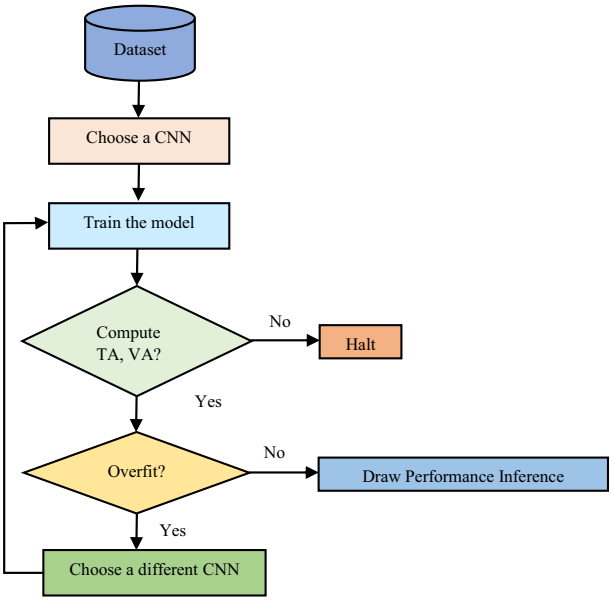


Fig.1 **a)** Layout of the proposed DRFEC for DR detection at an early stage **b)** Architectural Illustration of DRFEC



Flowchart 1 Working principle of DRFEC. Abbreviations: Training Accuracy (TA), Validation/Test Accuracy (VA)

images and validated using 7680 images. Figure 2 defines long tail distribution of the dataset. Figure 3 labels the fundus images (a) – (e) as left eye and (f) – (j) as right eye, with no DR, mild DR, moderate DR, severe DR and PDR, respectively. EyePACS [59] has 35,126 taken from different cameras with various FOV, with a resolution of 1440 × 960. It is available in jpeg format and is graded by many experts.

The noteworthy features are extracted and classified using 26 DL networks individually, in the DRFEC. The model is assessed and evaluated for classification, based on DR classes as 0, 1, 2, 3 and 4. The proposed model uses Adam optimizer, 0.0001 learning rate, categorical cross entropy loss function and softmax activation, for 50 epochs with a batch size of 10. The annotations of the images in the dataset are depicted in Table 1. Table 2 demonstrates the DR clinical manifestations corresponding to the category of DR in fundus images.

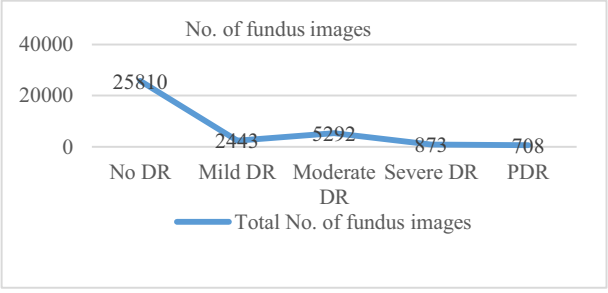


Fig. 2 Long tail distribution of the skewed DR dataset

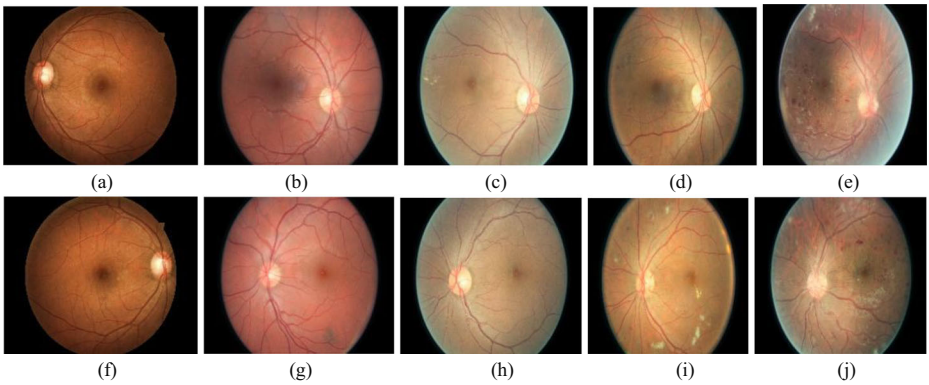


Fig. 3 Fundus of left eye (**a**) – (**e**) and right eye (**f**) – (**j**) depicting Grade 0, Grade 1, Grade 2, Grade 3 and Grade 4 DR, respectively

3.3 Image pre-processing

The image preprocessing module is the transparent module in CNN after the process of image dataset acquisition. The fundus images are raw high-resolution images and requires downsampling with respect to the compatibility standards of different CNN architectures such as for VGG-16 the image is down-sampled to (224 × 224 × 3). This helps the model to better interpret the image. In addition to these, the detection of bright intensity structures such as EXs and Optic Disc (OD) is ambiguous as EXs refer to a lesion whereas OD refers to normal anatomy of the retina. An abnormal OD can lead to subtle lesions which remains undetected due to its bright intensity. Therefore, image preprocessing techniques are important for identification and differentiation of artefacts from lesions for better detection of the disease. Additionally, identification of lesions leading to intermediate stages of DR, is also important for early DR detection. In the proposed model, the CNN's inbuilt pre-processing is used to extract image specifics which performs convolution and pooling, image striding and padding, using different filters and kernels. However, these DL CNN models are black box in nature and enhancement on images are implemented in real-time. Therefore, the enhanced image output cannot be visualized and only gradient details and edges can be extracted during feature extraction for classification.

3.4 NN-DL for feature extraction and classification

CNNs are convolutions with non-linear activation functions. CNN-DL feature extraction and classification is the subsequent component of the proposed DRFEC. The DL models perform feature extraction using numerous layers of convolution and pooling, and normalization and

Table 1 Number of labelled images

Stage of DR	No. of fundus images
No DR	25,810
Mild DR	2443
Moderate DR	5292
Severe DR	873
PDR	708

Table 2 DR clinical manifestations corresponding to the category

Category	Level	Clinical Manifestation
No DR	0	No DR lesions
Mild DR	1	MAs or HEs
Moderate DR	2	MAs, soft EXs, HEs, Venous Beading
Severe DR	3	Severe HEs, Venous Beading, mild IRMA
Proliferative DR (PDR)	4	Neovascularization

activation operations. In the meantime, such models also have the competence to learn and cause novel features from pre-existing features using representation learning, such as lines, boundary, points, edges, corners, vascular structure, etc., for simplification in detection. For instance, CNNs can perform edge detection using Prewitt Gradient Operator for the image [[3,0,1,2,7,4],[1,5,8,9,3,1],[2,7,2,5,1,3],[0,1,3,1,7,8],[4,2,1,6,2,8],[2,4,5,2,3,9]] as shown in Figs. 4 and 5(a) depicts Horizontal Prewitt Gradient kernel and Fig. 5(b) depicts Vertical Prewitt Gradient kernel.

The image array is convolved with the given Vertical Prewitt Gradient kernel of size of 3 × 3, which divides the 6 × 6 image into 16 regions of size 3 × 3, for detection of vertical edges. The formula for the convolution operation is shown in Fig. 6.

Thus, for the purpose of deep feature extraction from DR fundus images, state-of-the-art DL models such as VGG-19, VGG-16, Xception, InceptionV3, InceptionResNetV2, MobileNet, MobileNetV2, EfficientNet B0-B7, DenseNet121, DenseNet169, DenseNet201, ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2, NASNetLarge and NASNetMobile are used, to extract better and enhanced features necessary for the purpose of better learning and image classification. Figure 7 demonstrates the working of CNN which takes a grayscale input image (28x28x1) and performs convolution and pooling for feature extraction, and fully connected neural network-based classification using ReLU activation, dropout and softmax activation [64]. Figure 8 demonstrates the working of the CNN as a feature extractor which takes an RGB image (224x224x3) as an input and performs

Fig. 4 Image Array

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

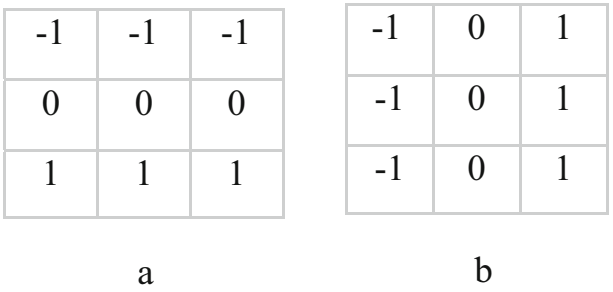


Fig. 5 Prewitt Gradient Kernel (**a**). Horizontal (**b**). Vertical

5 convolution and 5 pooling operations to enhance the representation of the images for better feature extraction and FCNN based classification.

Convolutional Networks are successful to a great extent in large-scale image classification using high performance computing and Graphical Processing Unit (GPU), thus making transitions from high-dimensional low feature encoding to deeper CNNs. These networks have undergone huge transformations, explored different connectivity patterns, and extracted multi-level features using skip-connections, inception modules, and dense blocks. Besides, cross-layer connections and architectural innovations contains small multi-layer perceptron in the kernels of convolutional layers to mine complex features. Additionally, the inner layers are supervised using auxiliary classifiers, to reinforce gradients expected from previous layers of intensely supervised networks, to expand the course of information by joining transitional stratums of dissimilar base networks, or growth of nets with paths that minimalize reconstruction losses.

4 Results

The proposed model is a composition of DL CNN models such as VGG-16, VGG-19, Xception, InceptionV3, InceptionResNetV2, MobileNet, MobileNetV2, EfficientNet B0-B7, DenseNet121, DenseNet169, DenseNet201, ResNet50,, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2, NASNetLarge and NASNetMobile which are trained on a training dataset of 27,446 fundus images and tested on a test dataset of 7680 fundus images, where both the training and test dataset contains images belonging to 5 classes

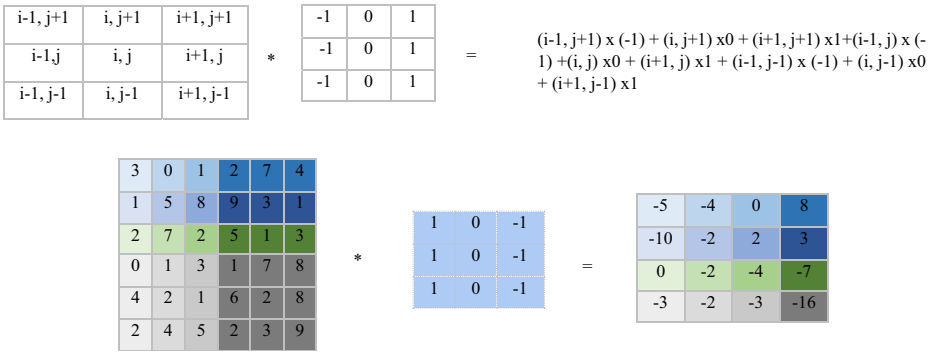


Fig. 6 Convolution of image array with Prewitt kernel for detection of vertical edges

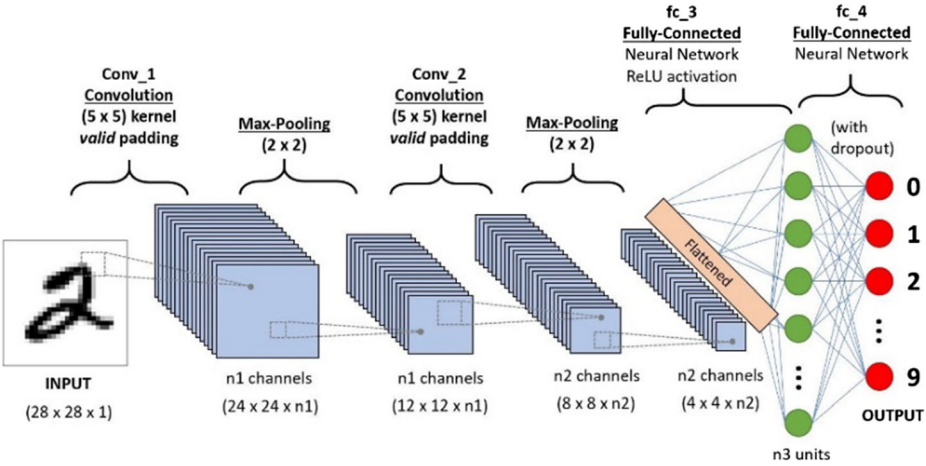
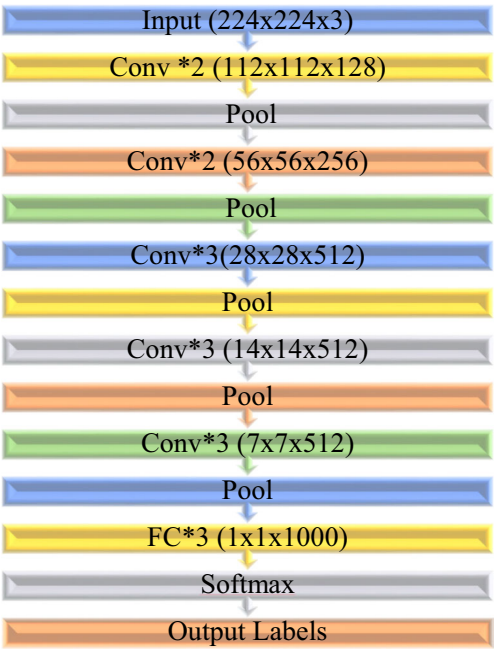


Fig. 7 Convolutional Neural Network [64]

of DR. The DL models are individually trained and tested upon the dataset of interest to evaluate their performance. The proposed model illustrates and demonstrates the behaviour of these models in a chronological fashion, and has achieved different training and validation values. The DL models are trained for a target size of (224 × 224), using inbuilt preprocessing of the CNNs, a batch size of 32, and using the fully connected classification layers for classification accompanied with softmax activation function. Besides, the DL models uses the state-of-the-art Adam optimizer, a learning rate of 0.0001 and categorical cross entropy loss function, for training for 50 epochs.

Fig. 8 CNN feature extractor



4.1 VGG-16

The VGG-16 model has a total trainable parameter of 138,357,544 and 0 non-trainable parameters. On regularization, through minimization of two layers of the neural network, the model has a total trainable parameter of 134,281,029 and 0 non-trainable parameters, for a target size of (224 × 224), using the intermediate layers for output. It has achieved a training accuracy of 90.91% and validation accuracy of 62.07%, in the 50th epoch. It has also achieved a training loss of 0.26 and a validation loss of 2.32. It can be inferred that the training accuracy and validation loss are higher than validation accuracy and training loss, respectively. This implies that the model is complex and is overfitting. To regularize the model and prevent it from overfitting, the layers of the network are reduced and the number of neurons are minimized or dropped to reduce model parameters. Besides, other forms of regularization such as preprocessing, data augmentation and batch normalization can also be applied to regularize the model. Figure 9 depicts training and validation loss of VGG-16. Figure 10 depicts training and validation accuracy of VGG-16.

4.2 VGG-19

The VGG-19 model has a total trainable parameter of 143,667,240 and 0 non-trainable parameter. On regularization, through minimization of two layers of the neural network, the model has a total trainable parameter of 139,590,725 and 0 non-trainable parameter, for a target size of (224 × 224). It has achieved a training accuracy of 97.98% and validation accuracy of 73.37%, in the 50th epoch. It has also achieved a training loss of 0.062 and a validation loss of 2.07. It can be inferred that the training accuracy and validation loss are higher than validation accuracy and training loss, respectively. This implies that the model is complex and is overfitting. It has many parameters that are capable of memorizing the training data, and hence are only capable of extracting information and not create it. Figure 11 depicts training and validation loss of VGG-19. Figure 12 depicts training and validation accuracy of VGG-19.

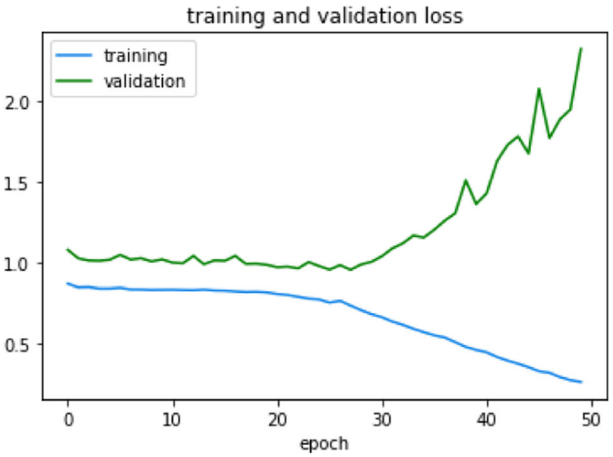


Fig. 9 Training and validation loss of VGG-16

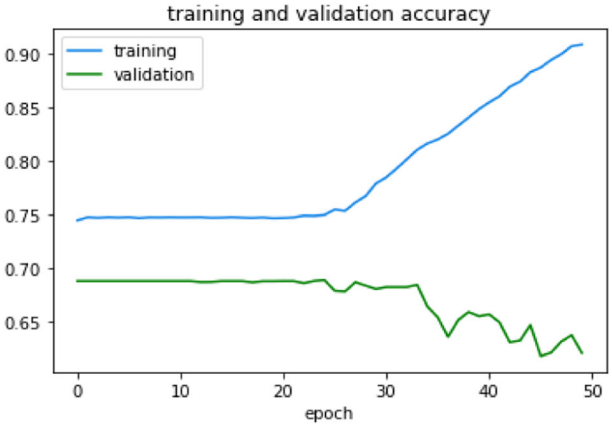


Fig. 10 Training and validation accuracy of VGG-16

4.3 ResNet101

The ResNet101 model has a total of 44,707,176 of which 44,601,832 parameters are trainable and 105,344 parameters are non-trainable, for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 42,668,421 of which 42,563,077 parameters are trainable and 105,344 parameters are non-trainable, for a target size of (224 × 224), upon the intermediate layers for output. It has achieved a training accuracy of 99.32% and a validation accuracy of 77.32%. It has also achieved a training loss of 0.0197 and a validation loss of 1.8714. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss. Figure 13 depicts training and validation loss of ResNet101. Figure 14 depicts training and validation accuracy of ResNet101.

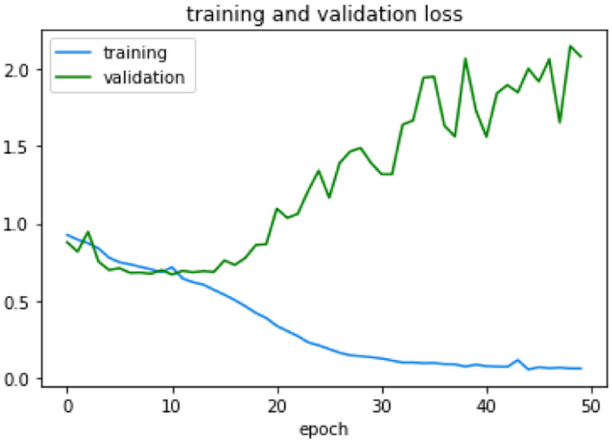


Fig. 11 Training and validation loss of VGG-19

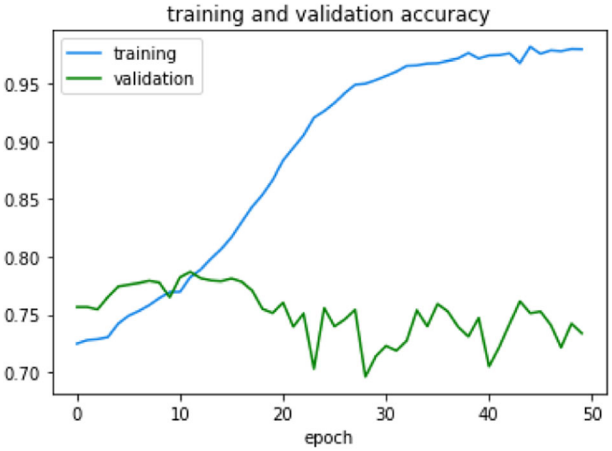


Fig. 12 Training and validation accuracy of VGG-19

4.4 ResNet101V2

The ResNet101V2 model has a total of 44,675,560 of which 44,577,896 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 42,636,805 of which 42,539,141 parameters are trainable. It has achieved a training accuracy of 99.34% and a validation accuracy of 75.40%. It has also achieved a training loss of 0.0197 and a validation loss of 1.9905. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss. Figure 15 depicts training and validation loss of ResNet101V2. Figure 16 depicts training and validation accuracy of ResNet101V2.

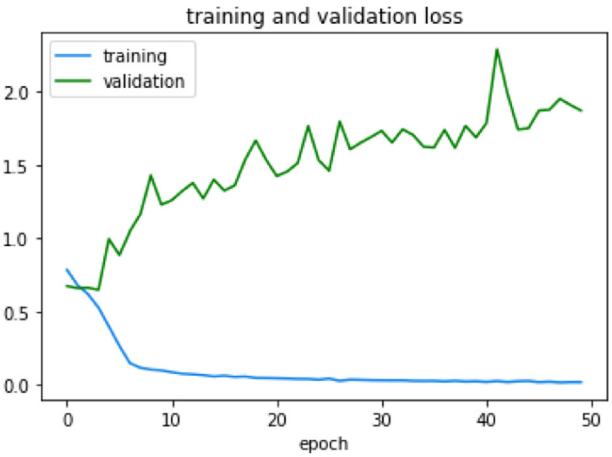


Fig. 13 Training and Validation loss of ResNet101

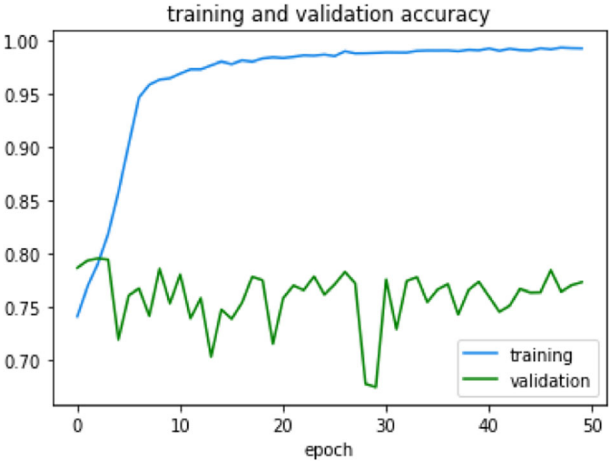


Fig. 14 Training and Validation accuracy of ResNet101

4.5 ResNet50

The ResNet50 model has a total of 25,636,712 of which 25,583,592 parameters are trainable and 53,120 parameters are non-trainable, for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 23,597,957 of which 23,544,837 parameters are trainable and 53,120 parameters are non-trainable, for a target size of (224 × 224), using the intermediate layers for output. It has achieved a training accuracy of 99.37% and a validation accuracy of 71.64%. It has also achieved a training loss of 0.0178 and a validation loss of 2.51. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 17 depicts

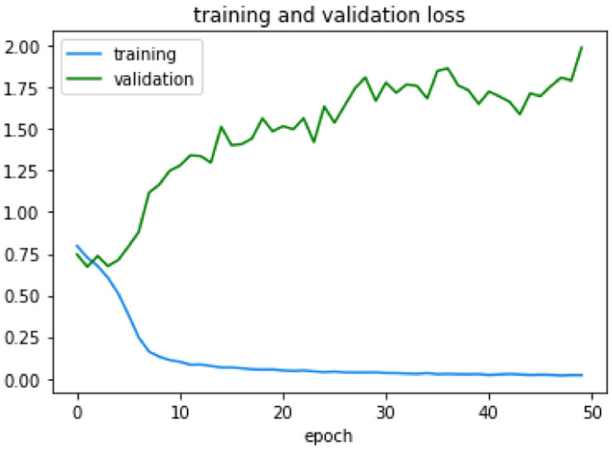


Fig. 15 Training and Validation loss of ResNet101V2

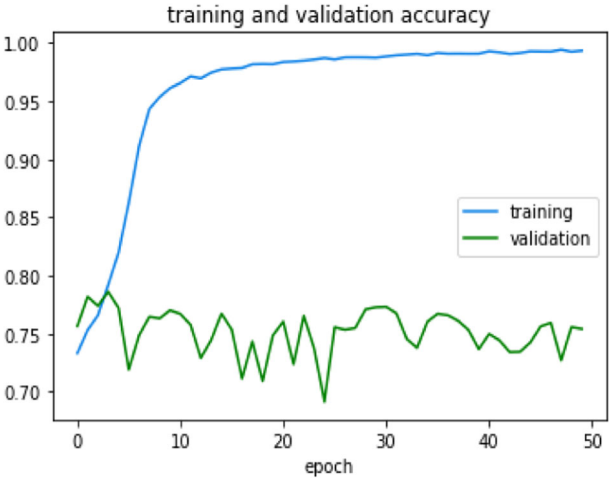


Fig. 16 Training and Validation accuracy of ResNet101V2

training and validation loss of ResNet50. Figure 18 depicts training and validation accuracy of ResNet50.

4.6 ResNet50V2

ResNet50 model has a total of 25,613,800 of which 25,568,360 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 23,575,045 of which 23,529,605 parameters are trainable. It has achieved a training accuracy of 99.19% and a validation accuracy of 72.97%. It has also achieved a training loss of 0.0238 and a validation loss of 1.8485. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal

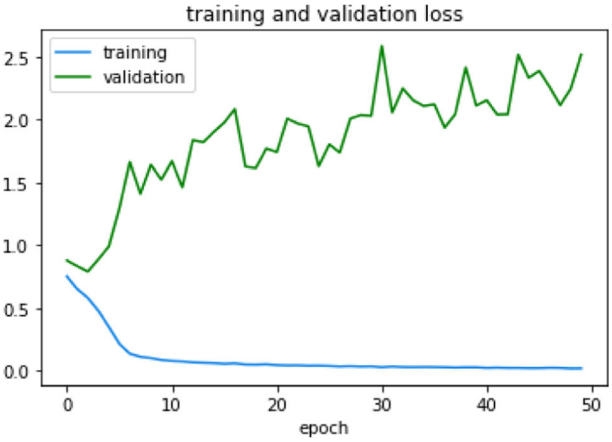


Fig. 17 Training and Validation loss of ResNet50

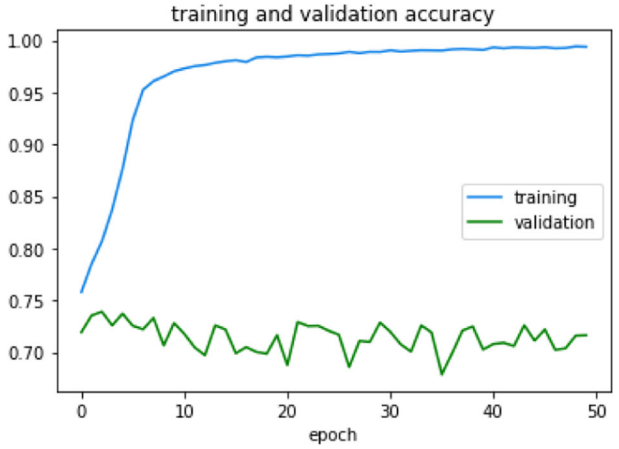


Fig. 18 Training and Validation accuracy of ResNet50

changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 19 depicts training and validation loss of ResNet50V2. Figure 20 depicts training and validation accuracy of ResNet50V2.

4.7 ResNet152

The ResNet152 model has a total of 60,419,944 of which 60,268,520 parameters are trainable and 151,424 parameters are non-trainable, for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 58,381,189 parameters of which 58,229,765 parameters are trainable and 151,424 parameters are non-trainable, for a target size of (224 × 224), using the intermediate layers for output. It has achieved a training accuracy of 99.44% and a validation accuracy of 76.65%. It has also achieved a training loss of 0.0163 and a validation loss of 1.81. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which

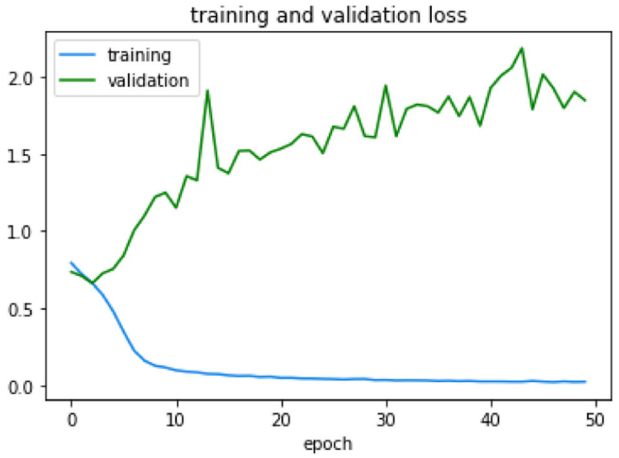


Fig. 19 Training and Validation loss of ResNet50V2

Fig. 20 Training and Validation accuracy of ResNet50V2

has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by two, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 21 depicts training and validation loss of ResNet152. Figure 22 depicts training and validation accuracy of ResNet152.

4.8 ResNet152V2

The ResNet152V2 model has a total of 60,380,648 of which 60,236,904 parameters are trainable, for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 58,341,893 parameters of which 58,198,149 parameters are trainable. It has achieved a training accuracy of 99.26% and a validation accuracy of 76.03%. It has also achieved a training loss of 0.0212 and a validation loss of 1.9876. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to

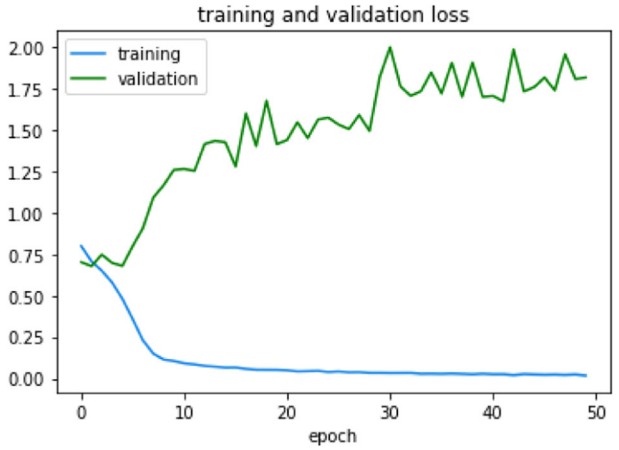


Fig. 21 Training and Validation loss of ResNet152

Fig. 22 Training and Validation accuracy of ResNet152

the overfitting of the model. On reducing and minimizing the number of layers in the neural network by two, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 23 depicts training and validation loss of ResNet152V2. Figure 24 depicts training and validation accuracy of ResNet152V2.

4.9 DenseNet121

The DenseNet121 model has a total of 8,062,504 parameters of which 7,978,856 parameters are trainable and 83,648 parameters are non-trainable, for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 7,042,629 parameters of which 6,958,981 parameters are trainable and 83,648 parameters are non-trainable, for a target size of (224 × 224), using the intermediate layers for output. It has achieved a training accuracy of 98.80% and a validation accuracy of 73.58%. It has also achieved a training loss of 0.0344 and a validation loss of 1.52. It is

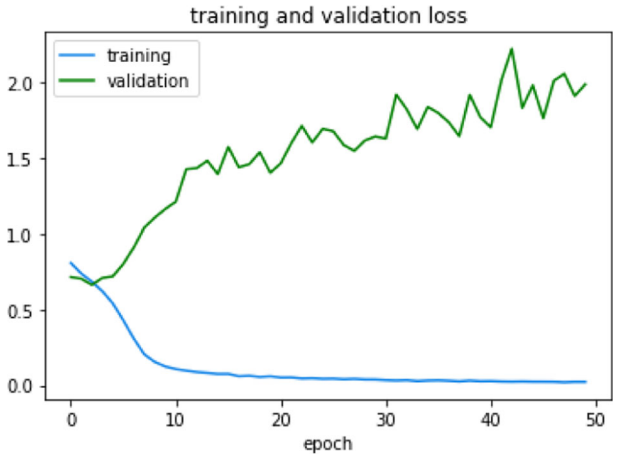


Fig. 23 Training and Validation loss of ResNet152V2

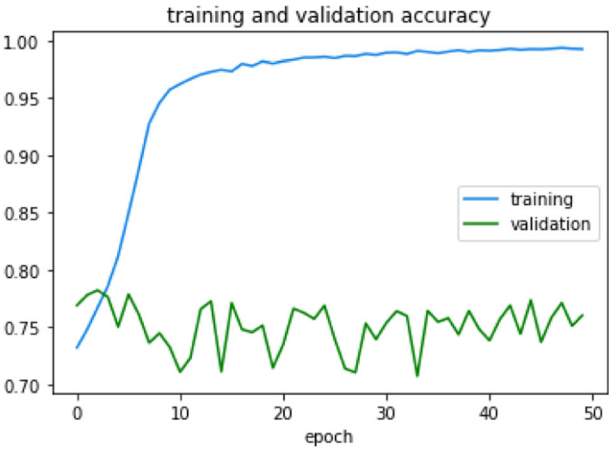


Fig. 24 Training and Validation accuracy of ResNet152V2

observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 25 depicts training and validation loss of DenseNet121. Figure 26 depicts training and validation accuracy of DenseNet121.

4.10 DenseNet169

The DenseNet169 model has a total of 14,307,880 parameters of which 14,149,480 parameters are trainable and 158,400 parameters are non-trainable, for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 12,651,205 parameters of which 12,492,805 parameters are trainable and 158,400 parameters are non-trainable, for a target size of (224 × 224), using

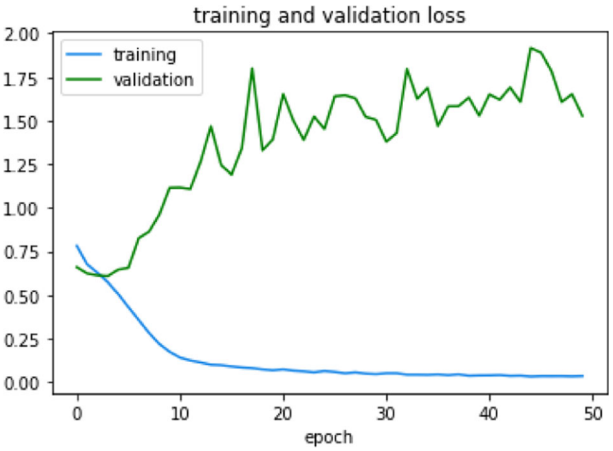


Fig. 25 Training and validation loss of DenseNet121

Fig. 26 Training and validation accuracy of DenseNet121

the intermediate layers for output. It has achieved a training accuracy of 98.95% and a validation accuracy of 79.02%. It has also achieved a training loss of 0.0314 and a validation loss of 1.94. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 27 depicts training and validation loss of DenseNet169. Figure 28 depicts training and validation accuracy of DenseNet169.

4.11 DenseNet201

The DenseNet201 model has a total of 20,242,984 parameters of which 20,013,928 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 18,331,589

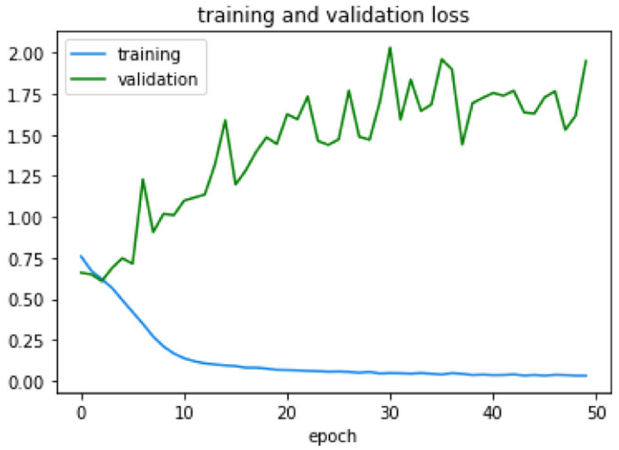


Fig. 27 Training and validation loss of DenseNet169

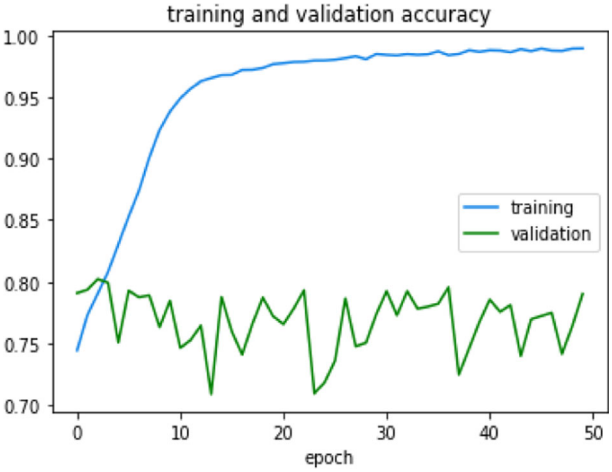


Fig. 28 Training and Validation accuracy of DenseNet169

parameters of which 18,102,533 parameters are trainable. It has achieved a training accuracy of 99.58% and a validation accuracy of 76.80%. It has also achieved a training loss of 0.0132 and a validation loss of 1.9230. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 29 depicts training and validation loss of DenseNet201. Figure 30 depicts training and validation accuracy of DenseNet201.

4.12 NASNetLarge

The NASNetLarge model has a total of 88,949,818 of which 88,753,150 parameters are trainable and 196,668 parameters are non-trainable, for a target size of (224 × 224). The

Fig. 29 Training and validation loss of DenseNet201

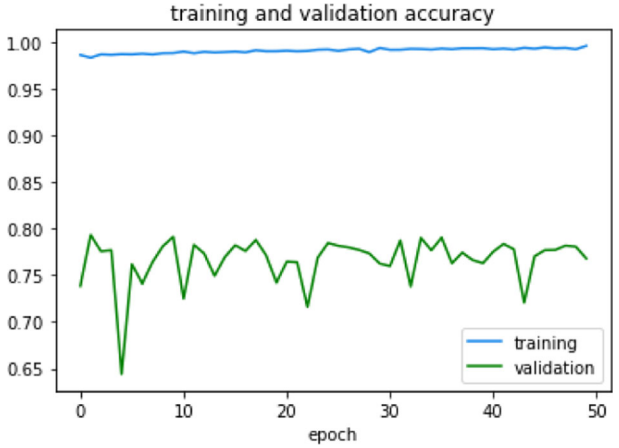


Fig. 30 Training and Validation accuracy of DenseNet201

model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 84,936,983 of which 84,740,315 parameters are trainable and 196,668 parameters are non-trainable, for a target size of (224 × 224), upon the intermediate layers for output. It has achieved a training accuracy of 99.24% and a validation accuracy of 79.09% It has also achieved a training loss of 0.0211 and a validation loss of 2.089. Figure 31 depicts training and validation loss of NASNetLarge. Figure 32 depicts training and validation accuracy of NASNetLarge.

4.13 NASNetMobile

The NASNetMobile model has a total of 5,326,716 of which 5,289,978 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 4,275,001 parameters of which 4,238,263 parameters are trainable. It has achieved a training accuracy of 99.05% and a validation accuracy of 67.49% It has also achieved a training loss of 0.0288 and

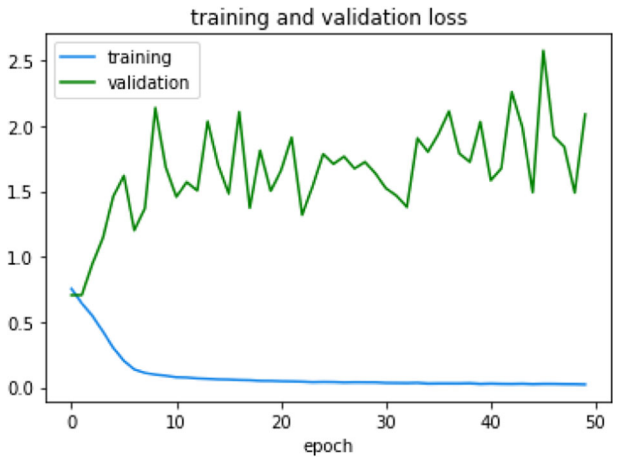


Fig. 31 Training and validation loss of NASNetLarge

Fig. 32 Training and validation accuracy of NASNetLarge

a validation loss of 1.7893. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 33 depicts training and validation loss of NASNetMobile. Figure 34 depicts training and validation accuracy of NASNetMobile.

4.14 Xception

The Xception model has a total of 22,910,480 parameters of which 22,855,952 parameters are trainable, for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 20,871,725 parameters of which 20,817,197 parameters are trainable. It has achieved a training accuracy

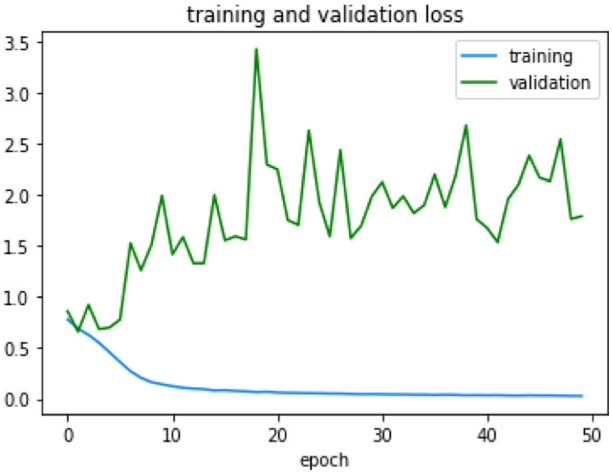


Fig. 33 Training and validation loss of NASNetMobile

Fig. 34 Training and Validation accuracy of NASNetMobile

of 99.46% and a validation accuracy of 78.05%. It has also achieved a training loss of 0.0175 and a validation loss of 1.88. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 35 depicts training and validation loss of Xception. Figure 36 depicts training and validation accuracy of Xception.

4.15 InceptionV3

The InceptionV3 model has a total of 23,851,784 parameters of which 23,817,352 parameters are trainable, for a target size of (224 × 224). The model is regularized by

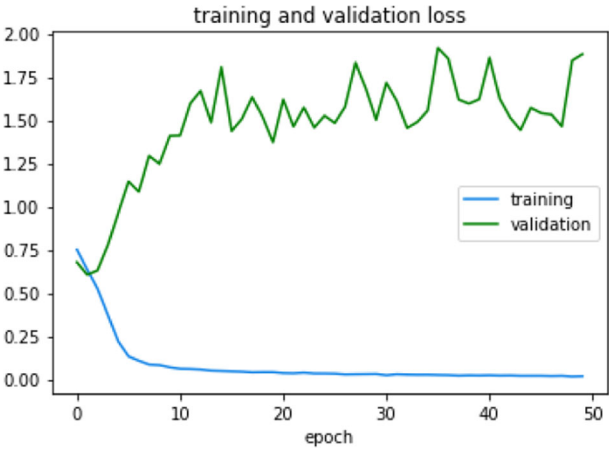


Fig. 35 Training and validation loss of Xception

Fig. 36 Training and validation accuracy of Xception

reducing the number of layers of the neural network by 2. This has led the model to have a total of 21,813,029 parameters of which 21,778,597 parameters are trainable. It has achieved a training accuracy of 99.03% and a validation accuracy of 73.72%. It has also achieved a training loss of 0.0265 and a validation loss of 1.51. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 37 depicts training and validation loss of InceptionV3. Figure 38 depicts training and validation accuracy of InceptionV3.

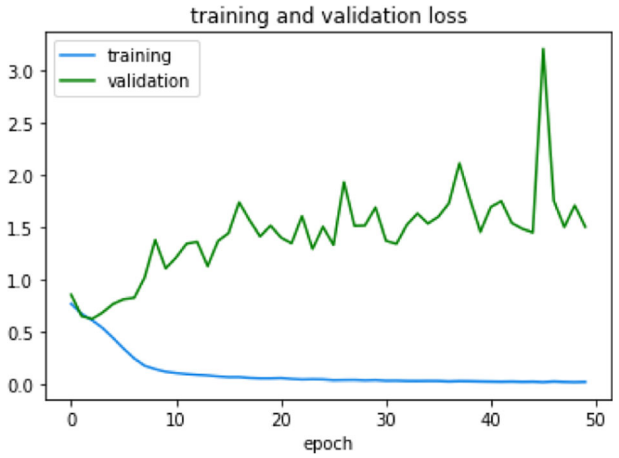


Fig. 37 Training and validation loss of InceptionV3

Fig. 38 Training and validation accuracy of InceptionV3

4.16 MobileNet

The MobileNet model has a total of 4,253,864 parameters of which 4,231,976 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 4,258,869 parameters of which 4,236,981 parameters are trainable. It has achieved a training accuracy of 98.92% and a validation accuracy of 77.20% It has also achieved a training loss of 0.0311 and a validation loss of 1.8153. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 39 depicts training and validation loss of MobileNet. Figure 40 depicts training and validation accuracy of MobileNet.

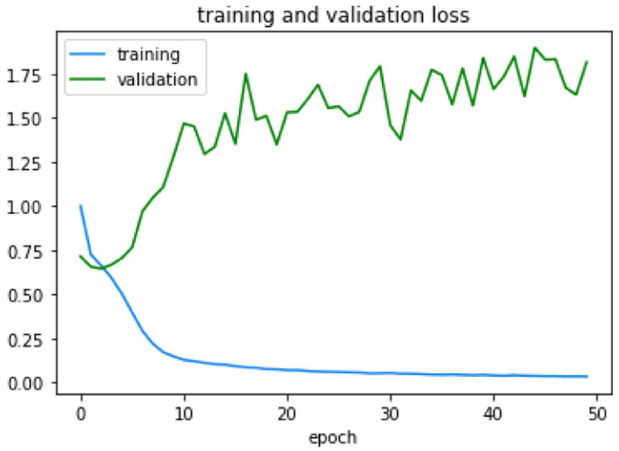


Fig. 39 Training and validation loss of MobileNet

Fig. 40 Training and validation accuracy of MobileNet

4.17 MobileNetV2

The MobileNetV2 model has a total of 3,538,984 parameters of which 3,504,872 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 2,264,389 parameters of which 2,230,277 parameters are trainable. It has achieved a training accuracy of 98.66% and a validation accuracy of 77.42%. It has also achieved a training loss of 0.0392 and a validation loss of 1.7161. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 41 depicts training and validation loss of MobileNetV2. Figure 42 depicts training and validation accuracy of MobileNetV2.

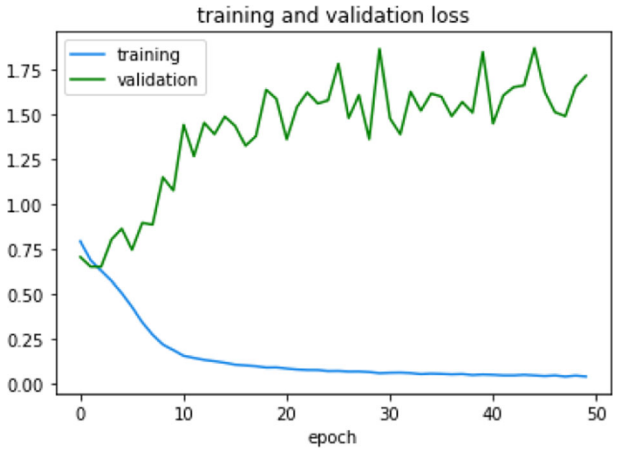


Fig. 41 Training and validation loss of MobileNetV2

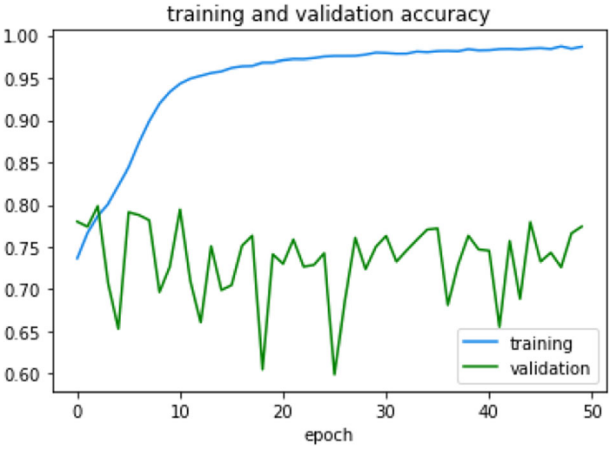


Fig. 42 Training and validation accuracy of MobileNetV2

4.18 InceptionResNetV2

The InceptionResNetV2 model has a total of 55,873,736 of which 55,813,192 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 54,344,421 parameters of which 54,283,877 parameters are trainable. It has achieved a training accuracy of 99.36% and a validation accuracy of 79.05%. It has also achieved a training loss of 0.0185 and a validation loss of 2.2305. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by two, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 43 depicts training and validation loss of InceptionResNetV2. Figure 44 depicts training and validation accuracy of InceptionResNetV2.

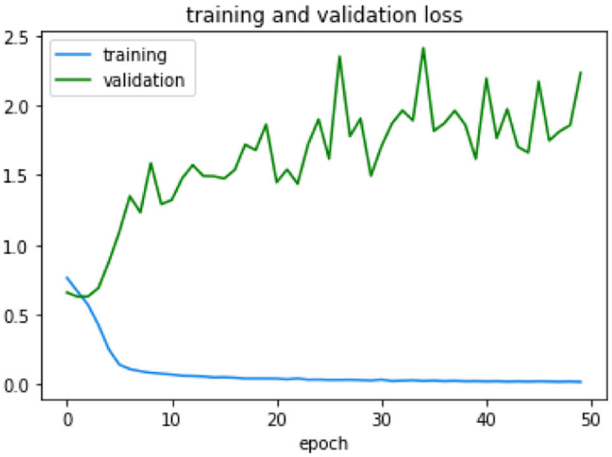


Fig. 43 Training and validation loss of InceptionResNetV2

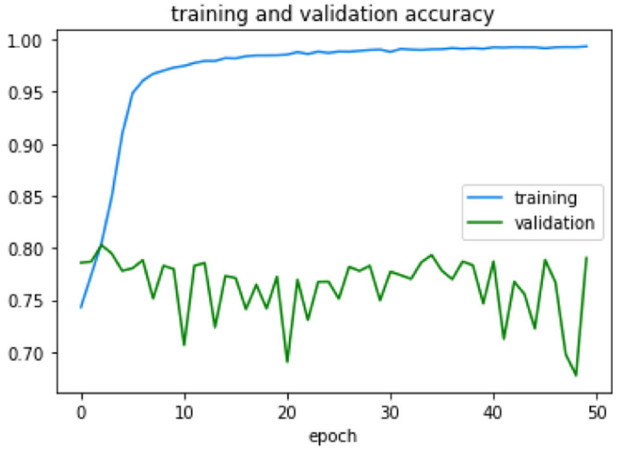


Fig. 44 Training and validation accuracy of InceptionResNetV2

4.19 EfficientNet B0

The EfficientNetB0 model has a total of 5,330,571 of which 5,288,548 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 4,055,976 of which 4,013,953 parameters are trainable. It has achieved a training accuracy of 99.03% and a validation accuracy of 75.63% It has also achieved a training loss of 0.03 and a validation loss of 1.88. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 45 depicts training and validation loss of EfficientNetB0. Figure 46 depicts training and validation accuracy of EfficientNetB0.

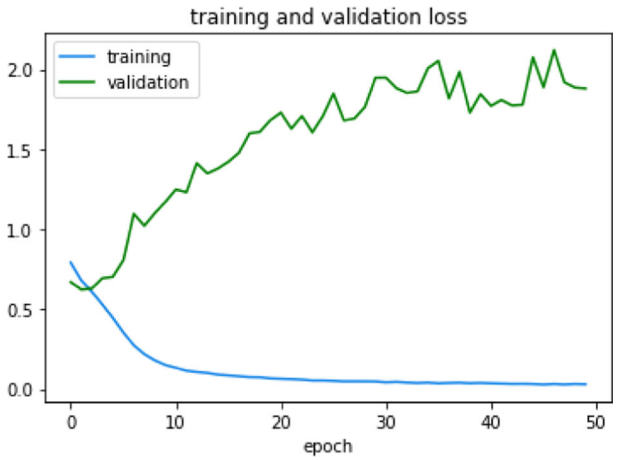


Fig. 45 Training and validation loss of EfficientNetB0

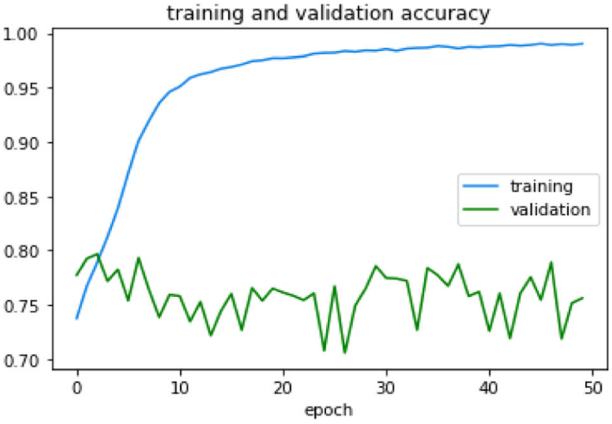


Fig. 46 Training and validation accuracy of EfficientNet B0

4.20 EfficientNet B1

The EfficientNetB1 model has a total of 7,856,239 of which 7,794,184 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 6,581,644 of which 6,519,589 parameters are trainable. It has achieved a training accuracy of 99.11% and a validation accuracy of 76.63% It has also achieved a training loss of 0.0268 and a validation loss of 1.7922. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 47 depicts training and validation loss of EfficientNetB1. Figure 48 depicts training and validation accuracy of EfficientNetB1.

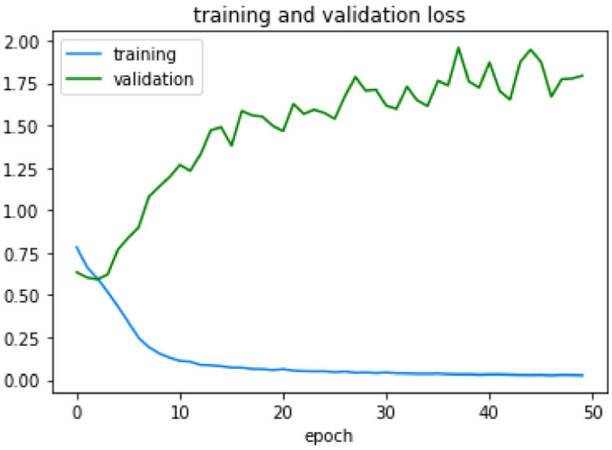


Fig. 47 Training and validation loss of EfficientNetB1

Fig. 48 Training and Validation accuracy of EfficientNetB1

4.21 EfficientNetB2

The EfficientNetB2 model has a total of 9,177,569 of which 9,109,994 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 7,775,614 of which 7,708,039 parameters are trainable. It has achieved a training accuracy of 99.20% and a validation accuracy of 76.35% It has also achieved a training loss of 0.0246 and a validation loss of 1.7861. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 49 depicts training and validation loss of EfficientNetB2. Figure 50 depicts training and validation accuracy of EfficientNetB2.

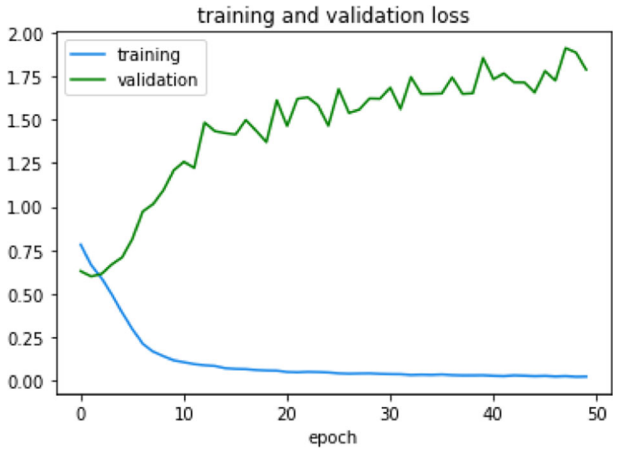


Fig. 49 Training and validation loss of EfficientNetB2

Fig. 50 Training and Validation accuracy of EfficientNetB2

4.22 EfficientNet B3

The EfficientNetB3 model has a total of 12,320,535 of which 12,233,232 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 10,791,220 of which 10,703,917 parameters are trainable. It has achieved a training accuracy of 99.20% and a validation accuracy of 78.14% It has also achieved a training loss of 0.0249 and a validation loss of 1.7087. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 51 depicts training and validation loss of EfficientNetB3. Figure 52 depicts training and validation accuracy of EfficientNetB3.

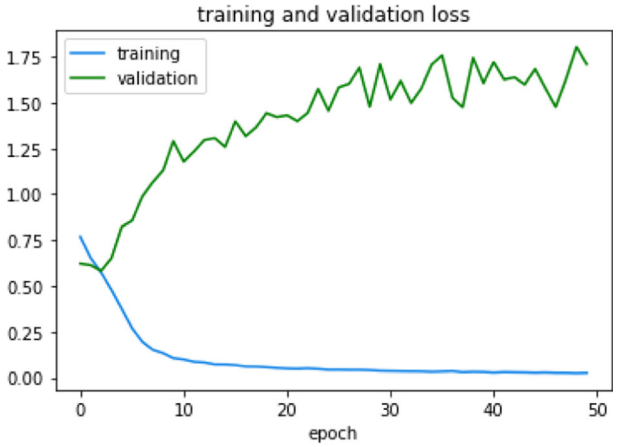


Fig. 51 Training and validation loss of EfficientNetB3

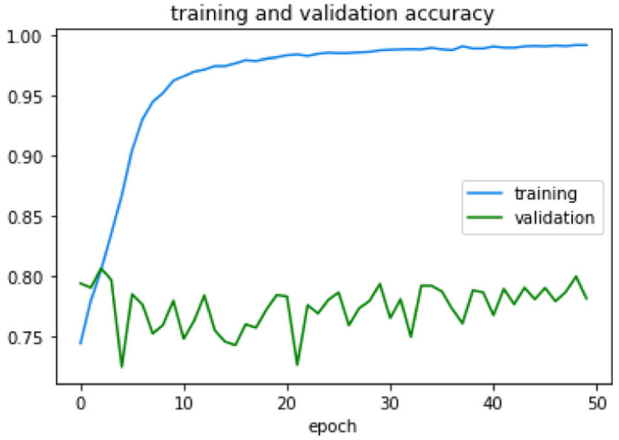


Fig. 52 Training and validation accuracy of EfficientNetB3

4.23 EfficientNet B4

The EfficientNetB4 model has a total of 19,466,823 of which 19,341,616 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 17,682,788 of which 17,557,581 parameters are trainable. It has achieved a training accuracy of 99.37% and a validation accuracy of 79.11% It has also achieved a training loss of 0.0192 and a validation loss of 1.5368. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 53 depicts training and validation loss of EfficientNetB4. Figure 54 depicts training and validation accuracy of EfficientNetB4.

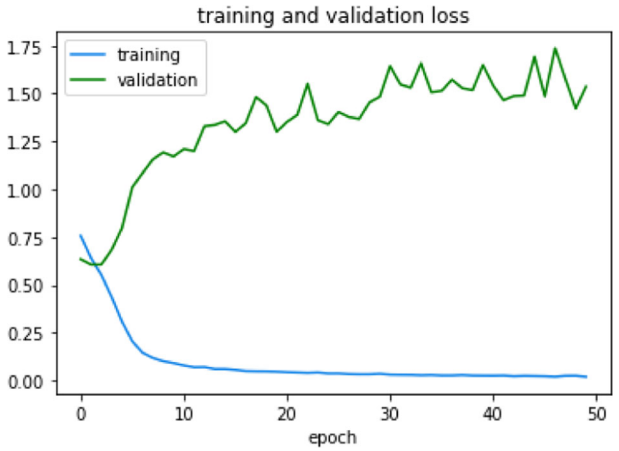


Fig. 53 Training and validation loss of EfficientNetB4

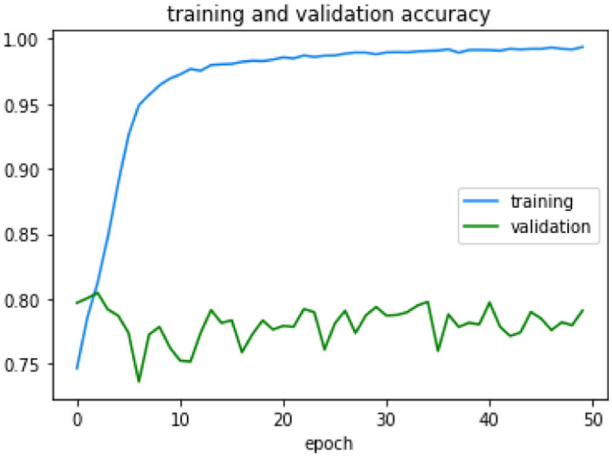


Fig. 54 Training and validation accuracy of EfficientNetB4

4.24 EfficientNet B5

The EfficientNetB5 model has a total of 30,562,527 of which 30,389,784 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 28,523,772 of which 28,351,029 parameters are trainable. It has achieved a training accuracy of 99.31% and a validation accuracy of 77.16% It has also achieved a training loss of 0.0196 and a validation loss of 1.5469. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 55 depicts training and validation loss of EfficientNetB5. Figure 56 depicts training and validation accuracy of EfficientNetB5.

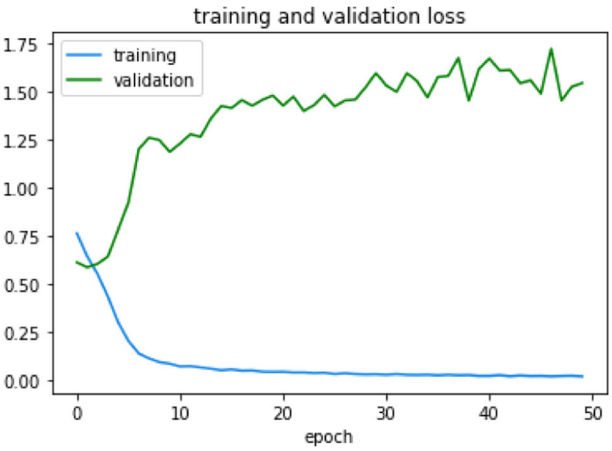


Fig. 55 Training and validation loss of EfficientNetB5

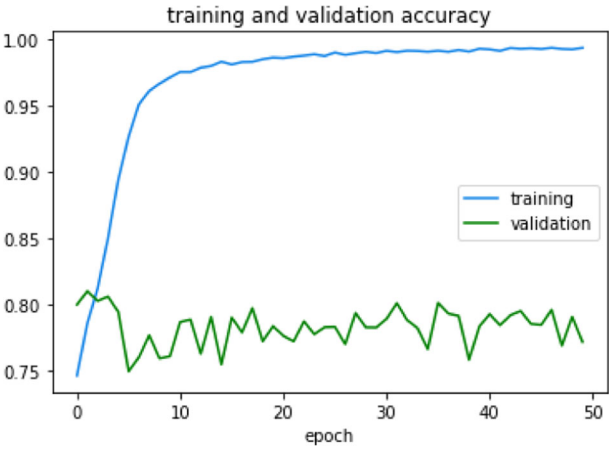


Fig. 56 Training and validation accuracy of EfficientNetB5

4.25 EfficientNet B6

The EfficientNetB6 model has a total of 43,265,143 of which 43,040,704 parameters are trainable for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 40,971,668 of which 40,747,229 parameters are trainable. It has achieved a training accuracy of 99.31% and a validation accuracy of 77.86% It has also achieved a training loss of 0.0197 and a validation loss of 1.5932. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 57 depicts training and validation loss of EfficientNetB6. Figure 58 depicts training and validation accuracy of EfficientNetB6.

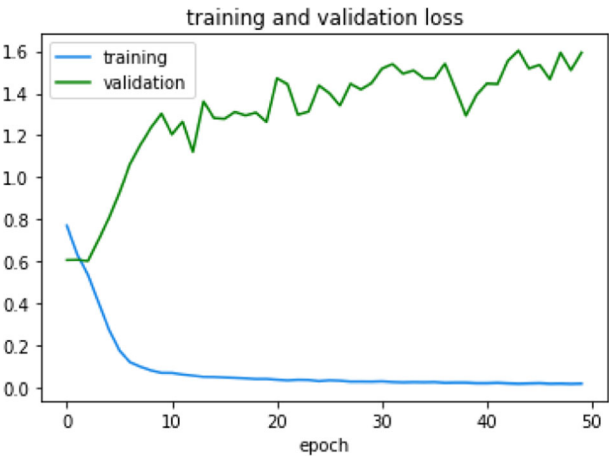


Fig. 57 Training and validation loss of EfficientNetB6

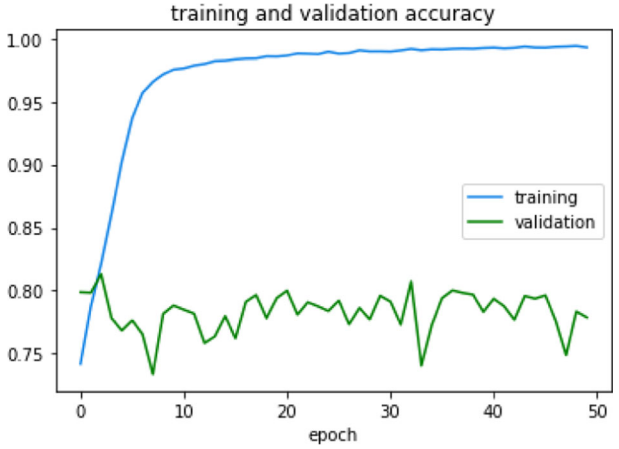


Fig. 58 Training and validation accuracy of EfficientNetB6

4.26 EfficientNet B7

The EfficientNetB7 model has a total of 66,658,687 of which 66,347,960 parameters are trainable and 310,727 parameters are non-trainable, for a target size of (224 × 224). The model is regularized by reducing the number of layers of the neural network by 2. This has led the model to have a total of 64,110,492 of which 63,799,765 parameters are trainable and 310,727 parameters are non-trainable, for a target size of (224 × 224), upon the intermediate layers for output. It has achieved a training accuracy of 99.35% and a validation accuracy of 72.97% It has also achieved a training loss of 0.0181 and a validation loss of 1.96. It is observed that the training accuracy and validation loss of the model are higher than validation accuracy and training loss, respectively which has led to the overfitting of the model. On reducing and minimizing the number of layers in the neural network by 2, for the purpose of regularization to reduce the overfitting caused, has however brought minimal changes in the gap of training and validation loss, except for minimization in the number of parameters. Figure 59 depicts training and validation loss of EfficientNetB7. Figure 60 depicts training and validation accuracy of EfficientNetB7.

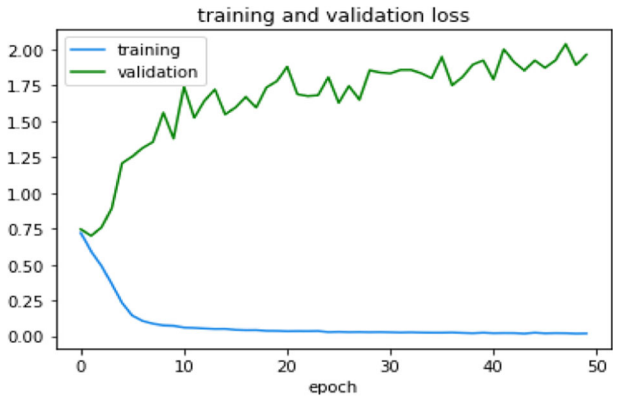


Fig. 59 Training and validation loss of EfficientNetB7

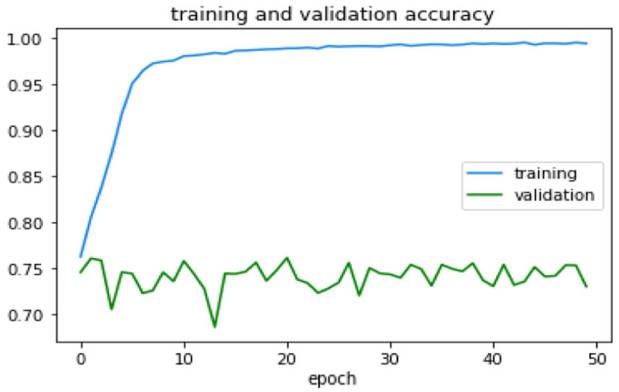


Fig. 60 Training and validation accuracy of EfficientNetB7

On the basis of the performance of the 26 DL models in training and testing accuracy, the proposed model is significant and uses Kolmogorov-Smirnov (KS) [27, 45] testing principle to determine the critical value D_{crit} and maximum absolute difference. On comparison, the proposed model has achieved a D_{crit} of 1 which is greater than 0.05, and a maximum absolute difference of 37.72. On this basis we reject our null hypothesis which claims on choosing and finding a best DL model for image classification, and accept our alternative hypothesis of finding and engineering a better DL model for DR image classification. Figures 61 and 62 depicts the plot of KSPA statistical test for 26 DL models, and comparison of them with contemporary and state-of-the-art works [1, 16, 18, 19, 23, 70, 72, 82], respectively.

5 Time complexity of DRFEC

The proposed model is a comprehensive evaluation of 26 CNN Deep Learning networks. The ML models especially Neural Networks (NNs) when applied on image data requires processing of thousands of features from thousands of pixels in a given image. Moreover, NNs are

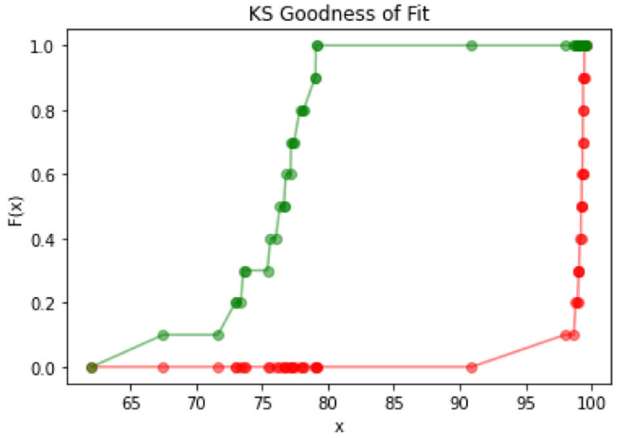


Fig. 61 KSPA statistical test for 26 DL models

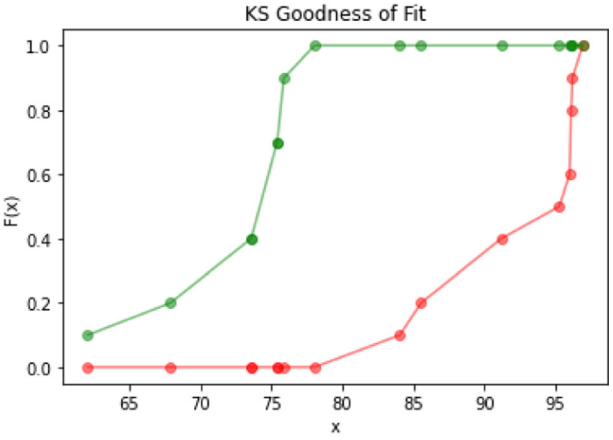


Fig. 62 KSPA statistical test for 26 DL models vs existing models

fully connected since inception and hence are highly parameterized. When they perform image learning, the number of parameters increases exponentially, as we will see in this section while analysing the working of a fully connected network in one of the most popular DL models. These networks cause overfitting and requires a large number of samples for proper training, in addition to larger memory and greater computational power for prediction. In image data, there are two major characteristics namely – feature localization or feature correlation and feature independence of location. CNNs thus, have the ability to solve the problem associated with too-many-parameters, with shared parameters (feature independence of location) of locally connected networks (feature localization) called the Convolution Net (ConvNet), which reduces the total number of parameters. These ConvNets have undergone evolution in its depth, architecture and search space. They are incorporated factors such as activation function, dropout, augmentation, filter size for parameter reduction and non-linear transformation, doubling of channels to recover lost information, to excel in performance. However, issues such as vanishing gradient deaccelerates their performance and makes them hard to optimize. They have introduced the concepts of variable filter sizes, residual learning, skip connections, identity mappings, depthwise separable convolutions and reinforcement learning to overcome problems associated with the depth of the network, and in the search space to identify the best combination of parameters.

In a CNN, the number of features is at most a constant time the number of input pixels say p, where the constant is <1. Therefore, the convolution of a fixed size filter across an image with p number of pixels takes *O(p)* time because each output is a sum of the product between *k pixels* in the *image* and *k weights* in the *filter,* where k does not vary with p. Similar is the case with maxpooling and average pooling operations of a CNN which does not take more than liner time in the given input size. Therefore, the overall runtime is linear [86]. With increasing depth, these CNNs transform to a Deep Neural Network (DNN). With the increase in the number of fully connected layers DNNs processes a feed-forward pass algorithm and a backpropagation algorithm [46]. Theoretically and mathematically, the training of a neural network-based model is implemented using matrices, and the time complexity of matrix multiplication ($M_{ij}*M_{jk}$) is O (i∗j∗k). For e.g., to compute the time complexity of the forward pass algorithm for VGG-16 with 16 layers and 5 convolutional blocks, let ***i*** denote the number of nodes of the *input convolutional layer*, ***j*** denotes the number of nodes in the *second*

convolutional layer, ***k*** denotes the number of nodes in the *third convolutional layer*, ***l*** denotes the number of nodes in the *MaxPooling layer*, m denotes the number of nodes in the *dense layer* **1**, n denotes the number of nodes in *dense layer* **2** and o denotes the number of nodes in the *output layer*. Since there are 7 layers, 6 matrices are required to represent weights between the layers.

Let W_{ji}, W_{kj}, W_{lk}, W_{ml}, W_{nm}, W_{on} be the matrices where W_{ji} is a matrix with j rows and i columns and contains the weights from layer i to layer j, W_{kj} is a matrix with k rows and j columns and contains the weights from layer j to layer k. Similarly, W_{lk} is a matrix with l rows and k columns and contains the weights from layer k to layer l, W_{ml} is a matrix with m rows and l columns and contains the weights from layer l to layer m. Again, W_{nm} is a matrix with n rows and m columns and contains the weights from layer m to layer n, and W_{on} is a matrix with o rows and n columns and contains the weights from layer n to layer o. For t training examples (here t = 27,446), to propagate from layer i to j, the following matrix multiplication is performed $S_{jt} = W_{ji} * Z_{it}$ and has $O\ (j*i*t)$ time complexity. The activation function $Z_{jt} = f\ (S_{jt})$ has $O\ (j*t)$ time complexity because it is an element-wise linear operation. In total, from i $\rightarrow$ j there is a time complexity of

$$O\ (j*i*t + j*t) = O\ (j*t*(i+1)) \approx O\ (j*i*t) \tag{i}$$

Similarly, from j $\rightarrow$ k there is a time complexity of $O\ (k*j*t)$, from k $\rightarrow$ l there is a time complexity of $O\ (l*k*t)$, from l $\rightarrow$ m there is a time complexity of $O\ (m*l*t)$, from m $\rightarrow$ n there is a time complexity of $O\ (n*m*t)$, and from n $\rightarrow$ o there is a time complexity of $O\ (o*n*t)$. In total, the time complexity for feedforward propagation will be

$$\underbrace{O(((j*i*t) + (k*j*t) + (l*k*t) + (m*l*t))}_{\text{5-block iteration say for VGG-16}} + \underbrace{(n*m*t) + (o*n*t))}_{\text{Dense Layers}} \tag{ii}$$

$$= O(t*((ij+jk+kl+lm) + mn+no))$$

Thus, the time complexity of the DNN during forward propagation is.

$$O(p) + O(t*((ij + jk + kl + lm) + mn + no)) \tag{iii}$$

This is repeated for all the convolution operations in a CNN- here VGG-16 with 5 convolutional blocks, and thus the total time-complexity for forward propagation in VGG-16 is.

$$\begin{aligned} &5*(O(p) + O(t*((ij + jk + kl + lm) + mn + no))) \\ &= O(p) + O(t*((ij + jk + kl + lm) + mn + no)) \end{aligned} \tag{iv}$$

which is linear. The similar concept is followed across all the other DL models depending upon the number of layers i.e., the depth of the network. The structures such as inception modules, skip connections, dense blocks in InceptionV3, ResNets and DenseNets, respectively helps in minimization of parameters and optimizes the depth of the network thereby optimizing the computational time. Thus, all the DL models have a time complexity which is linear and is

only affected by the number of parameters in the network. In case, where the input size and the size of the kernel is same, the time complexity is *O (p)* [87].

The backpropagation algorithm proceeds as follows: Initially, the error E_O, from the output layer o to n is computed and the matrix containing the error for the nodes at layer o is represented as -.

$$E_{ot} = f^{`}(S_{ot}) \otimes (Z_{ot} - P_{ot}) \tag{v}$$

E_{ot} has o rows and t columns meaning each column is an error for the training example t, and ⊗ is the element-wise operation. The small change 'delta, D' in the weights called 'delta weights' between the layer o and n is computed as -.

$$D_{on} = E_{ot} * Z_{tn}, \text{where } Z_{tn} \text{ is the transpose of } Z_{nt} \tag{vi}$$

The weights are adjusted as $W_{on} = W_{on} - D_{on}$, and for o to n the time complexity is.

$$O(ot + otn + on) = O(o*t*n) \tag{vii}$$

From layer n to m, $E_{nt} = f`(S_{nt}) \otimes (W_{no} * E_{ot})$, $D_{nm} = E_{nt} * Z_{tm}$ and $W_{nm} = W_{nm} - D_{nm}$, where W_{no} is the transpose of W_{on}. The time complexity for n to m is *O (nt + not + ntm + nm) = O (n*t (o + m))*. Similarly, for m to l, l to k, k to j and j to i, the time complexities are *O(m*t(n + l)), O(l*t(m + k)), O(k*t(l + j))* and *O(j*t(k + i))*, respectively. Therefore, the total complexity is.

$$O(otn + tn(o + m) + (tm(n + l) + tl(m + k) + kt(l + j) + tj(k + i)))$$
$$= O(t*(on + mn + (ml + lk + kj + ji))) \tag{viii}$$

which is the same as the feed-forward pass algorithm.

Therefore, the time complexity for one epoch will be.

$$O(t*((ij + jk + kl + lm) + mn + no)) \tag{ix}$$

In case for e = 50 number of epochs, the DNN will be showing a time complexity of -.

$$O(p) + O(e*t*((ij + jk + kl + lm) + mn + no))$$
$$= O(p) + O(50*t*((ij + jk + kl + lm) + mn + no)) \tag{x}$$

which is again linear. Here, the time complexity of the proposed model is based on the number of parameters in the DL networks. It is directly proportional to the number of parameters i.e. it increases and decreases with the increase and decrease in the total number of parameters in the network, respectively. However, it is observed that models with higher complexity have performed comparatively better than models with lower complexity such as NasNetLarge [2]. A similar thing is observed in the proposed model where larger architectures such as DensNet169, NasNetLarge, and EfficientNetB4 have shown significant performances despite huge architectural, space and computational complexity. Each of these models have taken a maximum of 72 hrs of execution on 50 epochs using 35,126 images in the available experimental environment.

The complexity of DL models is computed using multiply-accumulates (MACC) operation and Floating-Point Operations Per Second (FLOPs) [9] with a batch size of 32. The input and output to convolutional layers are not vectors but three-dimensional feature maps of size H ×

W × C where H is the height of the feature map, W the width, and C the number of channels at each location. The number of MACCs for a convolution layer with kernel size K is:

$$K \times K \times C_{in} \times H_{out} \times W_{out} \times C_{out} \quad \text{(xi)}$$

Each pixel in the output feature map is of size $H_{out} \times W_{out}$ and performs a dot product of the weights, and a K × K window of input values across all input channels, C_{in}. The layer has C_{out} different convolution kernels, therefore it is repeated C_{out} times to create all the output channels. For an e.g., a 3 × 3 convolution with 512 filters, on a 14 × 14 input feature map with 7 channels, the MACC is 3 × 3 × 7 × 14 × 14 × 512 = 6,322,176. The MACC-FLOPs obtained individually for each of the DL model is depicted in Table 3. Figures 63 and 64 illustrates the graphical presentation of the time complexity of 26 DL models in terms of MACCs and FLOPs, respectively.

6 Observations

The proposed work aims to show a comparative analysis of state-of-the-art DL models for early detection of DR using a fundus image dataset. On comparison, it is observed that the DL architectures belonging to different ensembles/family have shown different behaviours using the image data. Tables 4 and 5 highlights the comparative performance of the DL models in terms of training and validation accuracy, training and validation loss, and loss difference. It is observed that the validation loss of the architectures implemented in the proposed approach is

Table 3 MACCs and FLOPs of 26 DL models

Models	MACC	Models	FLOPs
DenseNet121	9,834,496	MobileNetV2	19,251,752,448
NASNetMobile	–	EfficientNetB0	25,250,817,155
DenseNet169	9,834,496	NASNetMobile	36,386,375,168
DenseNet201	9,834,496	MobileNet	36,401,180,160
InceptionV3	301,989,888	EfficientNetB1	44,863,292,419
VGG-16	462,422,016	EfficientNetB2	64,744,811,203
MobileNetV2	983,449,600	EfficientNetB3	118,910,253,571
EfficientNetB1	1,677,721,600	DenseNet121	181,468,425,728
ResNet50	2,517,630,976	DenseNet169	215,113,858,560
ResNet50V2	2,517,630,976	ResNet50V2	223,153,548,800
ResNet101	2,517,630,976	ResNet50	247,310,282,240
ResNet101V2	2,517,630,976	DenseNet201	274,733,309,440
ResNet152	2,517,630,976	EfficientNetB4	285,362,507,779
ResNet152V2	2,517,630,976	InceptionV3	366,430,995,968
MobileNet	2,517,630,976	ResNet101V2	460,844,887,552
EfficientNetB2	3,251,736,576	ResNet101	485,056,212,480
EfficientNetB0	3,933,798,400	Xception	535,288,550,144
EfficientNetB3	5,898,240,000	EfficientNetB5	665,398,641,155
VGG-19	10,070,523,904	ResNet152V2	698,561,916,416
InceptionResNetV2	13,086,228,480	ResNet152	722,840,677,888
EfficientNetB4	16,647,192,576	InceptionResNetV2	842,993,092,096
Xception	37,853,593,600	VGG-16	990,726,778,368
EfficientNetB5	53,084,160,000	EfficientNetB6	1,234,453,191,939
EfficientNetB6	110,841,053,184	VGG-19	1,257,123,606,016
NasNetLarge	–	NASNetLarge	1,525,754,037,120
EfficientNetB7	213,517,926,400	EfficientNetB7	2,440,337,381,379

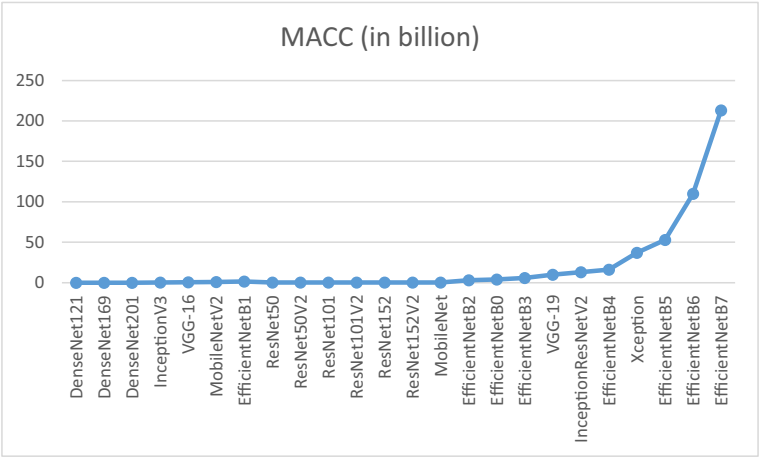


Fig. 63 Time complexity of 26 DL models in terms of MACCs

higher than the corresponding training loss. However, the training accuracy is significantly higher than the validation accuracy. This implies that the architectures inclines towards majority classification due to data imbalance which causes overfitting and poor generalization. Additionally, significant observations are analysed to identify the best DL architecture for early detection of DR which are discussed below:

i. In Table 4, it is shown that VGG-16 has the lowest training accuracy amongst all the DL architectures, due to its simple-deep architecture. However, VGG-19 has comparatively a better training and testing accuracy than VGG-16, due to increase in the number of layers. This implies that VGG can perform better with increasing depth. This difference in the behaviour of both VGG-16 and VGG-19 infers that the depth of the network has a crucial role to play in achieving better accuracy and minimum loss. However, with the increasing

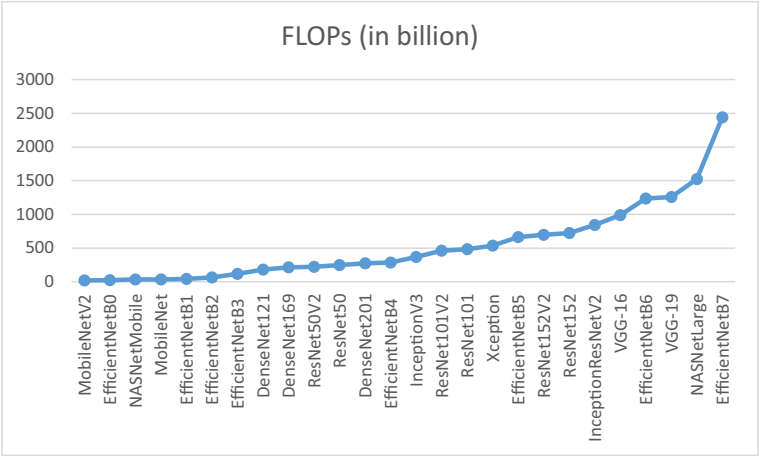


Fig. 64 Time complexity of 26 DL models in terms of FLOPs

Table 4 Performance comparison of CNNs in the DL model in terms of accuracy

Model	Training Accuracy (%)	Validation/Test Accuracy (%)	Performance Difference of DL n/w(s) w.r.t* EfficientNetB7 (%)
VGG-16	90.91	62.07	17.04
NASNetMobile	99.05	67.49	11.62
ResNet50	***99.37***	71.64	7.47
ResNet50V2	99.19	*72.97*	6.14
EfficientNetB7	**99.35**	*72.97*	6.14
VGG-19	97.98	73.37	5.74
DenseNet121	98.80	73.58	5.53
InceptionV3	99.03	73.72	5.39
ResNet101V2	**99.34**	75.40	3.71
EfficientNetB0	99.03	75.63	3.48
ResNet152V2	99.26	76.03	3.08
EfficientNetB2	99.20	76.35	2.76
EfficientNetB1	99.11	*76.63*	2.48
ResNet152	**99.44**	*76.65*	2.46
DenseNet201	**99.58**	76.80	2.31
EfficientNetB5	99.31	77.16	1.95
MobileNet	98.92	77.20	1.91
ResNet101	99.32	77.32	1.79
MobileNetV2	98.66	77.42	1.69
EfficientNetB6	99.31	77.86	1.25
Xception	**99.46**	78.05	1.06
EfficientNetB3	99.20	78.14	0.97
DenseNet169	98.95	79.02	0.09
InceptionResNetV2	**99.36**	79.05	0.06
NASNetLarge	99.24	79.09	0.02
EfficientNetB4	***99.37***	**79.11**	**0.00**

depth and number of parameters in the networks, performance issues such as vanishing gradient problem arises, which minimizes the effect of the loss function on the activation function. This causes VGG to perform poorly than the family of ResNets. The ResNet family has a better performance than VGG due to the initialization of skip connections in the network which not only retains the depth of the network but also eliminates the vanishing gradient problem.

ii. VGG-16 has achieved the lowest validation accuracy in comparison to all the other DL models. It imposes implicit regularization through greater depths and pre-initialization of network weights to avoid instability of gradients. It uses small convolutional filters for additional non-linearity in structure and captures spatial context based on non-trivial receptive fields. Thus, it is inferred that deeper models are beneficial for larger datasets, however VGG-16 has many weight parameters which makes the model heavy and causes larger inference time. This leads to vanishing gradient problem with higher loss and thus higher error with lesser ability to generalize, which ultimately affects validation performance. In addition to this, the model greatly suffers from overfitting as it is a simple yet overly deep network stacking 3 × 3 convolutional layers, and higher training error caused due to loss. VGG-19 is 11.3% better than VGG-16, 5.88% better than NasNetMobile, 1.73% better than ResNet50, and 0.40% better than ResNet50V2 and EfficientNetB7 respectively. Thus, VGGs are simple yet a better learning DL model than various state-of-the-art DL models.

Table 5 Performance comparison of CNNs in the DL model in terms of loss

Model	Training Loss (x)	Validation/Test Loss (y)	Loss Difference (z) z=y-x
ResNet50	0.017	2.51	2.493
VGG-16	0.26	2.32	2.06
InceptionResNetV2	0.018	2.23	2.212
NASNetLarge	0.021	2.09	2.069
VGG-19	0.062	2.07	2.008
ResNet101V2	0.019	1.99	1.971
ResNet152V2	0.021	1.98	1.959
EfficientNetB7	0.018	1.96	1.942
DenseNet169	0.031	1.94	1.909
DenseNet201	0.013	1.92	1.907
EfficientNetB0	0.03	1.88	1.85
Xception	0.017	1.88	1.863
ResNet101	0.019	1.87	1.851
ResNet50V2	0.024	1.84	1.816
MobileNet	0.031	1.81	1.779
ResNet152	0.016	1.8	1.784
NASNetMobile	0.029	1.79	1.761
EfficientNetB1	0.026	1.79	1.764
EfficientNetB2	0.024	1.78	1.756
MobileNetV2	0.039	1.72	1.681
EfficientNetB3	0.025	1.71	1.685
EfficientNetB6	0.019	1.59	1.571
EfficientNetB5	0.019	1.54	1.521
EfficientNetB4	0.019	1.53	1.511
DenseNet121	0.034	1.52	1.486
InceptionV3	0.026	1.51	1.484

iii. ResNet101 is 15.25% better than VGG-16, 9.83% better than NasNetMobile, 5.68% better than ResNet50, 4.35% better than ResNet50V2, 4.35% better than EfficientNetB7, 3.95% better than VGG-19, 3.74% better than ResNet121, 3.6% better than Inception V3, 1.92% better than ResNet101V2 and 1.69% better than EfficientNetB0. Moreover, it is 1.29% better than ResNet152V2, 0.97% better than EfficientNetB2, 0.69% better than EfficientNetB1, 0.67% better than ResNet152, 0.52% better than ResNet201, 0.16% better than EfficientNetB5 and 0.12% better than MobileNet. ResNets take lesser memory due to skip connections, faster inference time, and identity mapping leading to optimized training error.

iv. Amongst the inception-based modules which have better and optimized receptive fields of (1 × 1) and (3 × 3) instead of (5 × 5) and (7 × 7), to optimize the total number of parameters in the DL models of greater depth. This extreme inception module simplifies the learning task through explicit factoring at a smaller input space and maps all correlations using 1 × 1, 3 × 3 or 5 × 5 convolutions. It permits entire decoupling of spatial correlations and cross-channel correlations in the feature maps forming rich representations with fewer parameters. It has performed better with a training accuracy of 99.46% which is less than DenseNet201 having a training accuracy of 99.58%, by 0.12%. However, it is 15.98% better than VGG-16, 10.56% better than NasNetMobile, 6.41% better than ResNet50, and 5.08% better than ResNet50V2 and EfficientNetB7, respectively. Moreover, it is 4.68% better than VGG-19, 4.47% better than ResNet121, 4.33% better than Inception V3, 2.65% better than ResNet101V2 and 2.42% better than

EfficientNetB0. Additionally, it is 2.02% better than ResNet152V2, 1.7% better than EfficientNetB2, 1.42% better than EfficientNetB1, 1.4% better than ResNet152, 1.25% better than ResNet201, 0.89% better than EfficientNetB5, 0.85% better than MobileNet, 0.73% better than ResNet101, 0.63% better than MobileNetV2, and 0.19% better than EfficientNetB6. Xception is 0.10% better in training accuracy but 1% lower in testing accuracy, than InceptionResNetV2. It achieves performance gains due to the increase in the capacity of the network.

v. The family of DenseNets have especially a better performance than VGGs and ResNets because it reuses the features for learning, and eliminates the generation and learning of redundant features. It also has a better performance than various other DL models – it is 16.95% better than VGG-16, 11.53% better than NasNetMobile, 7.38% better than ResNet50, and 6.05% better than ResNet50V2 and EfficientNetB7, respectively. Moreover, it is 5.65% better than VGG-19, 5.44% better than ResNet121, 5.3% better than Inception V3, 3.62% better than ResNet101V2 and 3.39% better than EfficientNetB0. Additionally, it is 2.99% better than ResNet152V2, 2.67% better than EfficientNetB2, 2.39% better than EfficientNetB1, 2.37% better than ResNet152, 2.22% better than ResNet201, 1.86% better than EfficientNetB5, 1.82% better than MobileNet, 1.7% better than ResNet101, 1.6% better than MobileNetV2, 1.16% better than EfficientNetB6, 0.97% better than Xception and 0.88% better than EfficientNetB3.

vi. The family of NasNets is equipped with enriched computational power and automated engineering and solves the problem of finding a better CNN for image classification problem using Reinforcement Learning. It introduces a search space to find the best suited parameters, filter sizes, output channels, strides and total number of layers. On the basis of reinforcement learning it is able to determine the best searched architectures for the dataset. Despite the huge computational power required, NasNetLarge is 17.02% better than VGG-16, 11.60% better than NasNetMobile, 7.45% better than ResNet50, and 6.12% better than ResNet50V2 and EfficientNetB7, respectively. Moreover, it is 5.72% better than VGG-19, 5.51% better than ResNet121, 5.37% better than InceptionV3, 3.69% better than ResNet101V2 and 3.46% better than EfficientNetB0. Additionally, it is 3.06% better than ResNet152V2, 2.74% better than EfficientNetB2, 2.46% better than EfficientNetB1, 2.44% better than ResNet152, 2.29% better than ResNet201, 1.93% better than EfficientNetB5, 1.89% better than MobileNet, 1.77% better than ResNet101, 1.67% better than MobileNetV2, 1.23% better than EfficientNetB6, 1.04% better than Xception, 0.95% better than EfficientNetB3, 0.07% better than DenseNet169 and 0.04% better than InceptionResNetV2.

vii. DenseNet201 has achieved the highest training accuracy of 99.58% amongst all the DL models as well its corresponding performance-resembling architectures. It is observed that DenseNets have shown best results in representation of images, with strong gradient flow, better model complexity, diversified features, flexibility in feature complexity, smooth decision boundaries, and computational and parameter efficiency. However, the excessive connections of DenseNets can decrease its computational and parameter efficiency, and make it more prone to overfitting with highly skewed data. In brief, the depth of DenseNets have a crucial role to play in performance.

viii. EfficientNetB4 is 17.04% better than VGG-16, 11.62% better than NasNetMobile, 7.47% better than ResNet50, and 6.14% better than ResNet50V2 and EfficientNetB7, respectively. Moreover, it is 5.74% better than VGG-19, 5.53% better than ResNet121, 5.39% better than InceptionV3, 3.71% better than ResNet101V2 and 3.48% better than

EfficientNetB0. Additionally, it is 3.08% better than ResNet152V2, 2.76% better than EfficientNetB2, 2.48% better than EfficientNetB1, 2.46% better than ResNet152, 2.31% better than ResNet201, 1.95% better than EfficientNetB5, 1.91% better than MobileNet, 1.79% better than ResNet101, 1.69% better than MobileNetV2, 1.25% better than EfficientNetB6, 1.06% better than Xception, 0.97% better than EfficientNetB3, 0.09% better than DenseNet169, 0.06% better than InceptionResNetV2 and 0.02% better than NasNetLarge.

ix. EfficientNetB4 and ResNet50 achieved an optimal and equivalent training accuracy of 99.37%, but a testing accuracy 79.09% and 77.32%, respectively. Besides, ResNet50V2 and EfficientNetB7, and EfficientNetB1 and ResNet152 have the same testing accuracy of 72.97% and 76.6%, respectively. From this behaviour, it can be implied that the ResNets outperforms the EfficientNets in terms of training accuracy but the EfficientNets outperforms the ResNets in terms of validation accuracy. However, comparatively ResNets do not explicitly address generalization and suffers greatly from optimization difficulty. On the other hand, the fastidious nature of EfficientNets and the engineering of hyperparameters w.r.t the choice of ML frameworks for training affects their top-line accuracy. Additionally, they induce greater cost for data movement between layers due to larger number of channels, and have lesser applications because current hardware accelerators are engineered on traditional learning frameworks. From, Fig. 65 it is observed that the training curve increases in ResNet50 but decreases in EfficientNetB7 and further decreases in ResNet101V2. However, it increases in ResNet152 and then in DenseNet201 -the highest, and then decreases in Xception and then in InceptionResNetV2.

x. NASNetLarge and EfficientNetB4 have shown a near equivalent performance in training and validation accuracy. However, they are of different platforms. NASNet is based on reinforcement learning search method to optimize and find the best convolutional layer instead of searching for the entire network, and reduces computational complexity and improves generalization. It generates a search space that decouples complexity and depth of the network. However, EfficientNets such as EfficientNetB4 is a portable structure with depth and resembles MobileNets and NASNetMobile. From Tables 4 and 5, it is

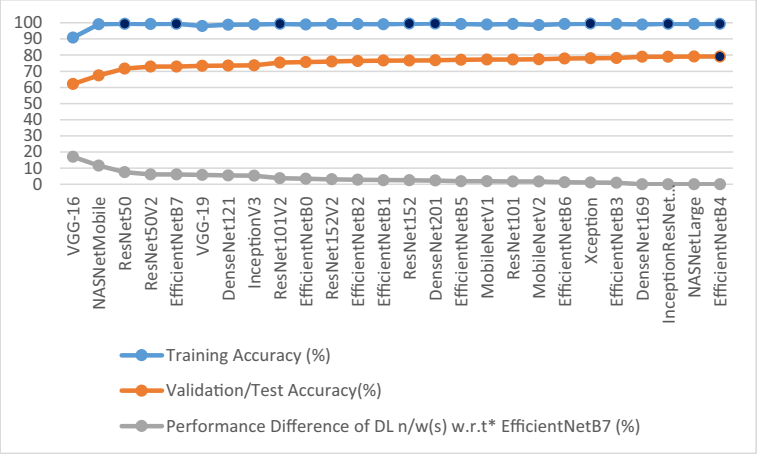


Fig. 65 Performance accuracy of 26 DL models w.r.t EfficientNetB7

observed that these portable, less memory-based structures have shown better performance due to computational and parameter efficiency than all other large DL models such as DenseNets, ResNets, VGGs, etc.

xi. EfficientNetB4 has achieved the highest validation accuracy amongst all the DL networks. Moreover, it has comparatively achieved a better training accuracy, training loss, validation loss and loss difference to that of DenseNet201 which has achieved the highest training accuracy. It can be inferred that EfficientNetB4 has shown remarkable performance compared to all the other DL models, despite overfitting and application of an imbalanced image dataset. ResNet50 and ResNet50V2 have shown highest overfitting while training the model and have achieved a highest loss difference of 2.49 each, whereas DenseNet121 and InceptionV3 have achieved a lowest loss difference of 1.48 each, thereby representing lowest overfitting.

Figure 65 is a graphical illustration of the accuracy of 26 DL models and their performance difference w.r.t* EfficientNetB7 in terms of accuracy.

Figure 66 is a graphical illustration of the accuracy of 26 DL models and their performance difference w.r.t* EfficientNetB7 in terms of loss.

7 Discussion

DRFEC is a competitive model and is able to achieve significant results. Dong et al. [23] have implemented an InceptionV3-VGG-16 based hybrid CNN through feature concatenation and have achieved 96.11% accuracy on 2693 images only whereas the proposed model has individually achieved an accuracy of 99.03% and 73.72% in InceptionV3 and 90.91% and 62.07% in VGG-16 using 35,126 images, which is significantly and comparatively larger to fit in a data hungry CNN. Deepa et al. [18] and Deepa et al. [19] have implemented a fine-tuned InceptionV3 and Xception based multi-stage patch-based and image-based CNN and have achieved an accuracy of 96.2% and 96%, respectively using a computationally powerful

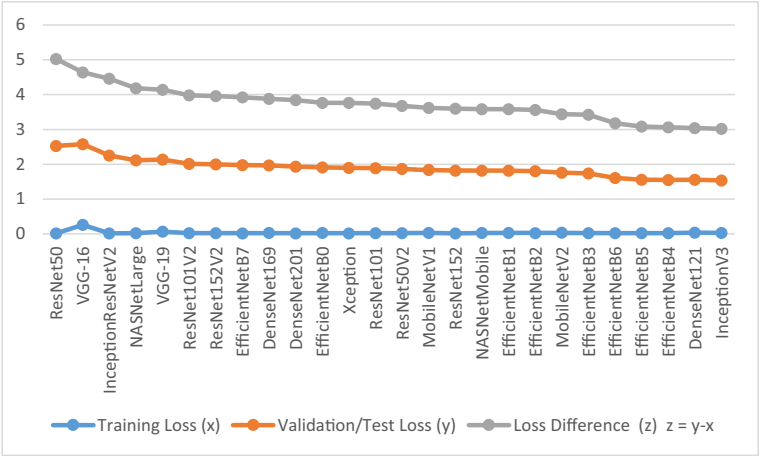


Fig. 66 Performance of 26 DL models w.r.t EfficientNetB7 in terms of loss

processing platform on various multi-sized image datasets, which are however overall, less than the Kaggle training dataset employed in proposed DRFEC. It has achieved comparatively an accuracy of 99.03% and 73.72% in InceptionV3 and 99.46% and 78.05% in Xception than Deepa et al. [18]. Tsai et al. [82] and AbdelMaksoud et al. [1] have implemented a DenseNet121 based CNN using Kaggle's EyePACs train-test dataset and TCH dataset, and using EyePACS Kaggle training dataset, IDRiD, MESSIDOR and APTOS 2019 datasets, and have achieved an accuracy of 84.05% and 84.67%, and 91.2% respectively. Whereas DRFEC has achieved an accuracy of 98.80% and 73.58% using DenseNet121 on 35,126 images. Das, S. et al. [16] have implemented Squeeze-and-Excitation CNN using limited DIARETDB1 and local dataset and achieved an accuracy of 96.92% whereas DRFEC has consistently achieved an accuracy of more than 98% and 70% on 35,126 images. Sau and Bansal [70] have implemented a FNU-GOA-MDNN for optimization using a comparatively limited ISBI 2018 IDRiD dataset and have achieved an accuracy of 95.27%, whereas DRFEC has consistently achieved an accuracy of more than 98% and 70% on 35,126 images. Shaik and Cherukuri [72] have implemented a VGG-16 based HA-Net using 3662 Kaggle's APTOS 2019 images and ISBI 2018 IDRiD images, and achieved an accuracy 85.54% and 66.41% respectively, whereas DRFEC has achieved an accuracy of 90.91% and 62.07% using VGG-16 and 97.98% and 73.37% using VGG-19 respectively on 35,126 images. It is observed the performance of the models, degrades with decreasing and limited data. The more is the data made available to a DL model the better it fits to be a better learning model which maintains its integrity and consistency, which is represented in DRFEC. Table 6 is an illustration which compares the performance of various existing and state-of-the-art DL models with the proposed DRFEC.

Conclusion This paper proposes an automated system for early detection of DR called Diabetic Retinopathy Feature Extraction and Classification (DRFEC) which employs DL CNN models such as VGG-16, VGG-19, Xception, InceptionV3, InceptionResNetV2, MobileNet, MobileNetV2, EfficientNets B0-B7, DenseNet121, DenseNet169, DenseNet201, ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2, NASNetLarge and NASNetMobile, for DR feature extraction and image classification. The proposed model performs an exhaustive analysis of these architectures upon fundus images, and derives the best performing DL architecture for DR feature extraction and fundus image classification. Amongst all the models, DenseNet201 has achieved the highest training accuracy whereas VGG-16 has achieved the lowest training accuracy. Again, VGG-16 has achieved lowest validation accuracy whereas EfficientNetB4 has achieved highest validation accuracy. The mobile-based structures such as EfficientNets and MobileNets have achieved better performance than residual networks, densely connected networks and inception modules. In the designed approach, the imbalanced dataset has caused overfitting of the models, but in addition to that the complexity of the DL models have also significantly contributed to overfitting, poor generalization, poor training time, poor gradient flow, and optimization and framework constraints. Moreover, ResNet50 has shown highest overfitting whereas InceptionV3 has shown the lowest overfitting. The proposed model has identified EfficientNetB4 and Xception as ideal DL models for DR deep feature extraction and image classification. EfficientNetB4 is the most optimal, efficient and reliable DL algorithm in detection of DR, followed by InceptionResNetV2, NasNetLarge and DenseNet169, and are the top-4 best performing models on Kaggle's EyePACS DR detection dataset. The experimental results show that DR detection using DL CNN architectures can achieve significant performances. In future, the best DL architecture determined through the proposed work can

Table 6 Performance comparison of DRFEC with state-of-the-art models

Model	Dataset images	Models	Accuracy (%)	Environment
Dong et al. [23]	2693	InceptionV3 -VGG-16 based CNN	96.11	Intel (R) Xeon (R) W-2245 CPU and NVIDIA GeForce RTX 2080 SUPER
Deepa et al. [18]	LFH, BH, DIARETDB, STARE, e-ophtha, ROC and Kaggle	Fine-tuned InceptionV3 and Xception based multi-stage patch-based and image-based CNN	96.2	GPU 25GB RAM
Tsai et al. [82]	Kaggle's EyePACs train and test dataset and TCH dataset	DenseNet121	84.05 and 84.67	–
Das, S. et al. [16]	DIARETDB1 and local dataset	Squeeze-and-Excitation CNN	96.92	–
AbdelMaksoud et al. [1]	EyePACS Kaggle training dataset, IDRiD, MESSIDOR and APTOS 2019	E-DenseNetBC-121	91.2	core i5/2.4 GHz, 8GB RAM, NVIDIA VGA card with 1GB VRAM.
Deepa et al. [19]	2290 (LFH, BH)	Fine-tuned Xception	96	GPU 25GB RAM
Sau and Bansal [70]	ISBI 2018 IDRiD	Optimized CNN FNU-GOA-MDNN	95.27	–
Shaik and Cherukuri [72]	3662 and ISBI 2018 IDRiD	VGG-16 based HA-Net	85.54 and 66.41	32GB RAM, 2 TB HDD enabled with 16GB NVIDIA GPU
Proposed DRFEC	Kaggle training dataset 35,126 splitted into train and test dataset	VGG-16	90.91 and 62.07	CPU 64GB RAM
		NASNetMobile	99.05 and 67.49	
		ResNet50	***99.37*** and 71.64	
		ResNet50V2	99.19 and 72.97	
		EfficientNetB7	**99.35** and 72.97	
		VGG-19	97.98 and 73.37	
		DenseNet121	98.80 and 73.58	
		InceptionV3	99.03 and 73.72	
		ResNet101V2	**99.34** and 75.40	
		EfficientNetB0	99.03 and 75.63	
		ResNet152V2	99.26 and 76.03	
		EfficientNetB2	99.20 and 76.35	
		EfficientNetB1	99.11 and 76.63	
		ResNet152	**99.44** and 76.65	
		DenseNet201	**99.58** and 76.80	

Table 6 (continued)

Model	Dataset images	Models	Accuracy (%)	Environment
		EfficientNetB5	99.31 and 77.16	
		MobileNet	98.92 and 77.20	
		ResNet101	99.32 and 77.32	
		MobileNetV2	98.66 and 77.42	
		EfficientNetB6	99.31 and 77.86	
		Xception	**99.46** and 78.05	
		EfficientNetB3	99.20 and 78.14	
		DenseNet169	98.95 and **79.02**	
		InceptionResNetV2	**99.36** and **79.05**	
		NASNetLarge	99.24 and **79.09**	
		EfficientNetB4	***99.37*** and **79.11**	

be used to incorporate state-of-the-art techniques such as attention mechanism for extraction of relevant features, for detection of subtle lesions, for early detection of DR. The proposed model will further be extended through architectural engineering, and using a balanced fundus image benchmark dataset for comparison with benchmark models and to overcome the limited resource constraint, to achieve more convincing results. The model also aims to mitigate overfitting and poor generalization in future works, through fine-tuning of DL models for computation of significant evaluation metrics such as AUROC. The proposed baseline model shall be used for comparison with newly proposed future model for better research direction.

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Data availability There is no additional data associated with this manuscript.

References

1. AbdelMaksoud, E, Barakat, S, Elmogy, M (2022) A computer-aided diagnosis system for detecting various diabetic retinopathy grades based on a hybrid deep learning technique, Med Biol Eng Comput, pp.1–24 https://doi.org/10.1007/s11517-022-02564-6
2. Agneeswaran, VS (n.d.) Computational Complexity of Deep Learning: Solution Approaches, Walmart Global Tech Blog, Available at:https://medium.com/walmartglobaltech/computational-complexity-of-deep-learning-a-birds-eye-view-2250b7c098a1, Accessed on: 20-05-2022
3. Atwany MZ, Sahyoun AH, Yaqub M (2022) Deep learning techniques for diabetic Retinopathy classification: a survey. IEEE Access 10:28642–28655. https://doi.org/10.1109/ACCESS.2022.3157632
4. Bhatti UA, Huang M, Wu D, Zhang Y, Mehmood A, Han H (2018) Recommendation system using feature extraction and pattern recognition in clinical care systems. Enterprise Inf Syst 13(3):329–351. https://doi.org/10.1080/17517575.2018.1557256
5. Bhatti UA, Huang M, Wang H, Zhang Y, Mehmood A, Di W (2018) Recommendation system for immunization coverage and monitoring. Human Vaccines Immunotherapeutics 14(1):165–171. https://doi.org/10.1080/21645515.2017.1379639
6. Bhatti UA, Yu Z, Li J, Nawaz SA, Mehmood A, Zhang K, Yuan L (2020) Hybrid watermarking algorithm using Clifford algebra with Arnold scrambling and chaotic encryption. IEEE Access 8:76386–76398. https://doi.org/10.1109/ACCESS.2020.2988298
7. Bhatti UA, Yu Z, Chanussot J, Zeeshan Z, Yuan L, Luo W, Nawaz SA, Bhatti MA, Ain Q, Mehmood A (2021) Local similarity-based spatial–spectral fusion hyperspectral image classification with deep CNN and Gabor filtering. IEEE Trans Geosci Remote Sens 60(5514215):1–15. https://doi.org/10.1109/TGRS.2021.3090410
8. Bhatti UA, Zeeshan Z, Nizamani MM, Bazai S, Yu Z, Yuan L (2022) Assessing the change of ambient air quality patterns in Jiangsu Province of China pre-to post-COVID-19. Chemosphere 288:1–10. https://doi.org/10.1016/j.chemosphere.2021.132569
9. Bhilare, A (2022) MACC-FLOPS, Available at: https://github.com/AbhijeetBhilare777/MACC-FLOPS, Accessed on : 06-06-2022
10. Bodapati JD, Veeranjaneyulu N, Shareef SN, Hakak S, Bilal M, Maddikunta PKR, Jo O (2020) Blended multi-modal deep convnet features for diabetic retinopathy severity prediction. Electronics 9(6):914. https://doi.org/10.3390/electronics9060914
11. Bora A, Balasubramanian S, Babenko B, Virmani S, Venugopalan S, Mitani A, Marinho GO, Cuadros J, Ruamviboonsuk P, Corrado GS, Peng L, Webster DR, Varadarajan AV, Hammel N, Liu Y, Bavishi P (2021) Predicting the risk of developing diabetic retinopathy using deep learning. Lancet Digit Health 2020(3):e10–e19. https://doi.org/10.1016/S2589-7500(20)30250-8
12. Chakraborty C, Kishor A, Rodrigues JJPC (2022) Novel enhanced-Grey wolf optimization hybrid machine learning technique for biomedical data computation. Comput Electr Eng 99:1–15. https://doi.org/10.1016/j.compeleceng.2022.107778
13. Chaturvedi SS, Gupta K, Ninawe V, Prasad PS (2020) Automated diabetic retinopathy grading using deep convolutional neural network, arXiv:2004.06334v1 [eess.IV], pp 1–12. https://doi.org/10.48550/arXiv.2004.06334

14. Chetoui, M, Akhloufi, MA (2020) Explainable Diabetic Retinopathy using EfficientNET, In 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp.1966–1969, https://doi.org/10.1109/EMBC44109.2020.9175664
15. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , arXiv:1610.02357v3 [cs.CV], pp 1251–1258. https://doi.org/10.48550/arXiv.1610.02357
16. Das S, Kharbanda K, Suchetha M, Raman R, Edwin DD (2021) Deep learning architecture based on segmented fundus image features for classification of diabetic retinopathy. Biomed Signal Process Control 68:1–10. https://doi.org/10.1016/j.bspc.2021.102600
17. Das D, Biswas SK, Bandyopadhyay S (2022) A critical review on diagnosis of diabetic retinopathy using machine learning and deep learning. Multimed Tools Appl 23:1–43. https://doi.org/10.1007/s11042-022-12642-4
18. Deepa V, Kumar CS, Cherian T (2021) Ensemble of multi-stage deep convolutional neural networks for automated grading of diabetic retinopathy using image patches, Journal of King Saud University –Computer and Information Sciences, pp 1–11 https://doi.org/10.1016/j.jksuci.2021.05.009
19. Deepa, V, Kumar, SC, Cherian, T (2022) Automated grading of diabetic retinopathy using CNN with hierarchical clustering of image patches by siamese network, Physical and Engineering Sciences in Medicine, pp.1–13 https://doi.org/10.1007/s13246-022-01129-z
20. Diabetic Retinopathy Detection, Kaggle (n.d.) Available at: https://www.kaggle.com/c/diabetic-retinopathy-detection/data. Accessed 01-07-2021
21. Diabetic Retinopathy Detection, Kaggle repository (n.d.) Available at: https://www.kaggle.com/c/diabetic-retinopathy-detection/data, Accessed on 19-06-2022
22. Diabetic Retinopathy, Updated (2022), Available at: https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/diabetic-retinopathy, Accessed on : 16-5-2022
23. Dong, B, Wang, X, Qiang, X, Du, F, Gao, L, Wu, Q, Cao, G, Dai, C (2022) A Multi-Branch Convolutional Neural Network for Screening and Staging of Diabetic Retinopathy Based on Wide-Field Optical Coherence Tomography Angiography, IRBM, pp.1–7 https://doi.org/10.1016/j.irbm.2022.04.004
24. Ege BM, Hejlesen OK, Larsen OV, Møller K, Jennings B, Kerr D, Cavan DA (2000) Screening for diabetic retinopathy using computer based image analysis and statistical classification, Comput. Methods Programs Biomed 62(3):165–175. https://doi.org/10.1016/S0169-2607(00)00065-1
25. Fadzil, MHA, Ngah, NF, George, TM, Izhar, LI, Nugroho, H, Nugroho, HA (2010) Analysis of foveal avascular zone in colour fundus images forgrading of diabetic retinopathy severity, 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology, pp. 5632–5635, https://doi.org/10.1109/IEMBS.2010.5628041
26. Fadzil MHA, Izhar LI, Nugroho H, Nugroho HA (2011) Analysis of retinal fundus images for grading of diabetic retinopathy severity. Med Biol Eng Compu 49(6):693–700. https://doi.org/10.1007/s11517-011-0734-2
27. Fan GF, Zhang LZ, Yu M, Hong WC, Dong SO (2022) Applications of random forest in multivariable response surface for short-term load forecasting. Int J Electr Power Energy Syst 139:1–30. https://doi.org/10.1016/j.ijepes.2022.108073
28. Fong DS, Aiello L, Gardner TW, King GL, Blankenship G, Cavallerano JD, Ferris FL, Klein R (2004) Retinopathy in diabetes. Diabetes Care 27(1):84–87. https://doi.org/10.2337/diacare.27.2007.S84
29. Goh JKH, Cheung CY, Sim SS, Tan PC, Tan GSW, Wong TY (2016) Retinal imaging techniques for diabetic retinopathy screening. J Diabetes Sci Technol 10(2):282–294. https://doi.org/10.1177/1932296816629491
30. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, Venugopalan S, Widner K, Madams T, Cuadros J, Kim R, Raman R, Nelson PC, Mega JL, Webster DR (2016) Development and validation of a deep learning algorithm for detection of diabetic Retinopathy in retinal fundus photographs. JAMA 316(22):2402–2410. https://doi.org/10.1001/jama.2016.17216
31. Gurcan OF, Beyca OF, Dogan O (2021) A Comprehensive Study of Machine Learning Methods on Diabetic Retinopathy Classification. Int J Comput Intell Syst 14(2):1132–1141. https://doi.org/10.2991/ijcis.d.210316.001
32. Hagos MT (2020) Point-of-care diabetic retinopathy diagnosis: a standalone mobile application approach, arXiv:2002.04066v1 [eess.IV]. Vol. abs/2002.04066, pp 1–84. Available at: https://www.semanticscholar.org/paper/Point-of-Care-Diabetic-Retinopathy-Diagnosis%3A-AHagos/3b9a121e53cea1b30cd29eea13854eeb4faadff9
33. Hattiya T, Dittakan K, Musikasuwan S (2021) Diabetic Retinopathy Detection Using Convolutional Neural Network: A Comparative Study on Different Architectures. Mahasarakham Int J Eng Technol 7(1):50–60. https://doi.org/10.14456/mijet.2021.8

34. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778. https://doi.org/10.1109/CVPR.2016.90
35. Howard, AG, Zhu, M, Chen, B, Kalenichenko, D, Wang, W, Weyand, T, Andreetto, M, Adam, H (2017) MobileNets: efficient convolutional neural networks for mobile vision applications, arXiv:1704.04861v1 [cs.CV], pp 1–9. https://doi.org/10.48550/arXiv.1704.04861
36. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700–4708, arXiv:1608.06993v5 [cs.CV]. https://doi.org/10.48550/arXiv.1608.06993
37. Huang G, Liu S, Maaten L, Weinberger KQ (2018) CondenseNet: an efficient densenet using learned group convolutionss. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2752–2761. https://doi.org/10.1109/CVPR.2018.00291
38. Hui J, Du M, Ye X, Qin Q, Sui J (2019) Effective building extraction from high-resolution remote sensing images with multitask driven deep neural network. IEEE Geosci Remote Sens Lett 16(5):786–790. https://doi.org/10.1109/LGRS.2018.2880986
39. Iandola, F, Mskewicz, M, Karayev, S, Girshick, R, Darrell, T, Kuetzer, K (2014) DenseNet: Implementing Efficient ConvNet Descriptor Pyramids Technical Report, pp.1–11 https://doi.org/10.1080/08839514.2013.848751
40. Islama MM, Yanga H, Poly TN, Jiane WS, Li YJ (2020) Deep learning algorithms for detection of diabetic retinopathy in retinal fundus photographs: a systematic review and meta-analysis. Comput Methods Prog Biomed 191(105320):1–16. https://doi.org/10.1016/j.cmpb.2020.105320
41. Janiesch C, Zschech P, Heinrich K (2021) Machine learning and deep learning. Electron Mark 31:685–695. https://doi.org/10.1007/s12525-021-00475-2
42. Ji Q, Huang J, He W, Sun Y (2019) Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images. Algorithms 12(3):1–12. https://doi.org/10.3390/a12030051
43. Jiang, H, Yang, K, Gao, M, Zhang, D, Ma, H, Qian, W (2019) An interpretable ensemble deep learning model for diabetic retinopathy disease classification, In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2045–2048, https://doi.org/10.1109/EMBC.2019.8857160
44. Kamal KC, Yin Z, Wu M, Wu Z (2021) Evaluation of deep learning-based approaches for COVID-19 classification based on chest X-ray images, signal. Image Vid Process 15:959–966. https://doi.org/10.1007/s11760-020-01820-2
45. Kawwa, N (2020) When to Use the Kolmogorov-Smirnov Test Theory, Application, and Interpretation, Towards Data Science, Available at: https://towardsdatascience.com/when-to-use-the-kolmogorov-smirnov-test-dd0b2c8a8f61, Accessed on: 12-06-2022
46. Kazem AM (2018) What is the time complexity for training a neural network using back-propagation?, ARTIFICIAL INTELLIGENCE, Stack Exchange, Available at: https://ai.stackexchange.com/questions/5728/what-is-the-time-complexity-for-training-a-neural-network-using-back-propagation, Updated on November 2021, Accessed on 20-05-2022
47. Kishor, A, Chakraborty, C (2021) Early and accurate prediction of diabetics based on FCBF feature selection and SMOTE, Int J Syst Assur Eng Manag, pp. 1–9, https://doi.org/10.1007/s13198-021-01174-z
48. Kishor, A, Chakraborty, C (2021) Artifcial Intelligence and Internet of Things Based Healthcare 4.0 Monitoring System, Wireless Personal Communications, pp.1–17 https://doi.org/10.1007/s11277-021-08708-5
49. Kishor, A, Jeberson, W (2021) Diagnosis of Heart Disease Using Internet of Things and Machine Learning Algorithms, In: Singh, P.K., Wierzchoń, S.T., Tanwar, S., Ganzha, M., Rodrigues, J.J.P.C. (eds) Proceedings of Second International Conference on Computing, Communications, and Cyber-Security, Lecture Notes in Networks and Systems, Vol 203, https://doi.org/10.1007/978-981-16-0733-2_49
50. Kishor A, Chakraborty C, Jeberson W (2020) A Novel Fog Computing Approach for Minimization of Latency in Healthcare using Machine Learning , Special Issue on Current Trends in Intelligent Multimedia Processing Systems. Int J Interact Multimed Artif Intell 6(7):7–17. https://doi.org/10.9781/ijimai.2020.12.004
51. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (Eds.) Adv Neural Inf Process Syst., Curran Associates Inc, pp 1097–1105. https://doi.org/10.1145/3065386
52. Kumar PNS, Deepak RU, Sathar A, Sahasranamam V, Kumar RR (2016) Automated detection system for diabetic retinopathy using two field fundus photography. Proc Comput Sci 93:486–494. https://doi.org/10.1016/j.procs.2016.07.237

53. Lee J, Kim YK, Park KH, Jeoung JW (2020) Diagnosing Glaucoma with spectral-domain optical coherence tomography using deep learning classifier. J Glaucoma 29(4):287–294. https://doi.org/10.1097/IJG.0000000000001458
54. Li N, Ma M, Lai M, Gu L, Kang M, Wang Z, Jiao S, Dang K, Deng J, Ding X, Zhen Q, Zhang A, Shen T, Zheng Z, Wang Y, Peng Y (2022) A stratified analysis of a deep learning algorithm in the diagnosis of diabetic retinopathy in a real-world study. J Diab 14(2):111–120. https://doi.org/10.1111/1753-0407.13241
55. Lim WX, Chen ZY, Ahmed A (2022) The adoption of deep learning interpretability techniques on diabetic retinopathy analysis: a review. Med Biol Eng Comput 60:633–642. https://doi.org/10.1007/s11517-021-02487-8
56. Mayyaa, V, Kamath, SS, Kulkarni, U (2021) Automated microaneurysms detection for early diagnosis of diabetic retinopathy: A Comprehensive review, Computer Methods and Programs in Biomedicine Update, 1, pp-1-15 https://doi.org/10.1016/j.cmpbup.2021.100013
57. Michele A, Colin V, Santika DD (2019) MobileNet convolutional neural networks and support vector Machines for Palmprint Recognition. Procedia Comput Sci 157:110–117. https://doi.org/10.1016/j.procs.2019.08.147
58. Nneji GU, Cai J, Deng J, Monday HK, Hossin MA, Nahar S (2022) Identification of diabetic Retinopathy using weighted fusion deep learning based on Dual-Channel fundus scans. Diagnostics 12(540):1–19. https://doi.org/10.3390/diagnostics12020540
59. Orlando JI, Prokofyeva E, del Fresno M, Blaschko MB (2018) An ensemble deep learning based approach for red lesion detection in fundus images. Comput Methods Prog Biomed 153:115–127. https://doi.org/10.1016/j.cmpb.2017.10.017
60. Padmanayana, Anoop BK (2022) Binary Classification of DR-Diabetic Retinopathy using CNN with Fundus Colour Images, Materials Today: Proceedings, pp.1–5 https://doi.org/10.1016/j.matpr.2022.01.466
61. Pogorelov, K, Riegler, M, Halvorsen, P , Griwodz, C, Lange, T, Randel, KR, Eskeland, SL, Dang-Nguyen, D , Ostroukhova, O, Lux, M, Spampinato, C (2017) A Comparison of Deep Learning with Global Features for Gastrointestinal Disease Detection, MediaEval'17, pp. 1–3 https://doi.org/10.1007/s11042-017-4989-y
62. Pour AM, Seyedarabi H, Jahromi SHA, Javadzadeh A (2020) Automatic detection and monitoring of diabetic Retinopathy using efficient convolutional neural networks and contrast limited adaptive histogram equalization. IEEE Access 8:136668–136673. https://doi.org/10.1109/ACCESS.2020.3005044
63. Priya R, Aruna P (2013) A new eyenet model for diagnosis of diabetic retinopathy. Appl Artif Intell 27(10):924–940. https://doi.org/10.1080/08839514.2013.848751
64. Ratan, P (2020) What is the Convolutional Neural Network Architecture?, Analytics Vidhya, Available at: https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/, Data Science Blogathon, October 28, 2020, Accessed on 19-05-2022
65. Saeed F, Hussain M, Aboalsamh HA (2021) Automatic diabetic Retinopathy diagnosis using adaptive fine-tuned convolutional neural network. IEEE Access 9:41344–41359. https://doi.org/10.1109/ACCESS.2021.3065273
66. Samanta A, Saha A, Satapathy SC, Fernandes SL, Zhang Y (2020) Automated detection of diabetic retinopathy using convolutional neural networks on a small dataset. Pattern Recogn Lett 135:293–298. https://doi.org/10.1016/j.patrec.2020.04.026
67. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4510–4520, arXiv:1801.04381v4 [cs.CV]. https://doi.org/10.48550/arXiv.1801.04381
68. Sarki, R, Michalska, S, Ahmed, K, Wang, H, Zhang, Y (2019) Convolutional neural networks for mild diabetic retinopathy detection: an experimental study, bioRxiv, pp.1–18 https://doi.org/10.1101/763136
69. Sarki R, Ahmed K, Wang H, Zhang Y, Ma J, Wang K (2021) Image preprocessing in classification and identification of diabetic eye diseases. Data Sci Eng 6:455–471. https://doi.org/10.1007/s41019-021-00167-z
70. Sau, PC, Bansal, A (2022) A novel diabetic retinopathy grading using modified deep neural network with segmentation of blood vessels and retinal abnormalities, Multimedia Tools and Applications, pp.1–29 https://doi.org/10.1007/s11042-022-13056-y
71. Shah P, Mishra DK, Shanmugam MP, Doshi B, Jayaraj H, Ramanjulu R (2020) Validation of Deep Convolutional Neural Network based algorithm for detection of diabetic retinopathy – Artificial intelligence versus clinician for screening. Indian J Ophthalmol 68(2):398–405. https://doi.org/10.1007/s11760-020-01820-2
72. Shaik, NS, Cherukuri, TK (2022) Hinge attention network: A joint model for diabetic retinopathy severity grading, Applied Intelligence, pp. 1–17 https://doi.org/10.1007/s10489-021-03043-5
73. Shukla, UV, Tripathy, K (2022) Diabetic Retinopathy, in: StatPearls [internet], Treasure Island (FL): StatPearls Publishing
74. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556v6 [cs.CV], pp 1–14. https://doi.org/10.48550/arXiv.1409.1556

75. Sivapriya, G, Praveen, V, Gowri, P, Saranya, S, Sweetha, S, Shekar, K (2022) Segmentation of Hard exudates for the detection of Diabetic Retinopathy with RNN based sematic features using fundus images, Materials Today: Proceedings, pp.1–9 https://doi.org/10.1016/j.matpr.2022.05.189
76. Sopharak A, Uyyanonvara B, Barman S (2009) Automatic exudate detection from non-dilated diabetic retinopathy retinal images using fuzzy c-means clustering. Sensors 9(3):2148–2161. https://doi.org/10.3390/s90302148
77. Sosale B, Aravind SR, Murthy H, Narayana S, Sharma U, Gowda SGV, Naveenam M (2020) Simple, Mobile-based Artificial intelligence algorithm in the detection of diabetic Retinopathy (SMART) study. BMJ Open Diabetes Res Care 8(1):1–6. https://doi.org/10.1136/bmjdrc-2019-000892
78. Suriyal, S, Druzgalski, C, Gautam, K (2018) Mobile assisted diabetic retinopathy detection using deep neural network, 2018 Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges (GMEPE/PAHCE), pp. 1–4, https://doi.org/10.1109/GMEPE-PAHCE.2018.8400760
79. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2818–2826. https://doi.org/10.1109/CVPR.2016.308
80. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), pp 1–7. https://doi.org/10.1609/aaai.v31i1.11231
81. Tan M, Le QV (2019) EfficientNet: rethinking model scaling for convolutional neural networksm. In: Proceedings of the 36th International Conference on Machine Learning, vol 97, pp 6105–6114, arXiv: 1905.11946v5 [cs.LG]. https://doi.org/10.48550/arXiv.1905.11946
82. Tsai CY, Chen CT, Chen GA, Yeh CF, Kuo CT, Hsiao YC, Hu HY, Tsai IL, Wang CH, Chen JR, Huang SC, Lu TC, Woung LC (2022) Necessity of local modification for deep learning algorithms to predict diabetic Retinopathy. Int J Environ Res Public Health 19(3):1–12. https://doi.org/10.3390/ijerph19031204
83. Tymchenko B, Marchenko P, Spodarets D (2020) Deep learning approach to diabetic retinopathy detection, arXiv:2003.02261v1 [cs.LG], pp 1–9. https://doi.org/10.48550/arXiv.2003.02261
84. Walter T, Klein JC, Massin P, Erginay A (2002) A contribution of image processing to the diagnosis of diabetic retinopathy-detection of exudates incolor fundus images of the human retina. IEEE Trans Med Imaging 21(10):1236–1243. https://doi.org/10.1109/TMI.2002.806290
85. Wang J, Liu Q, Xie H, Yang Z, Zhou H (2021) Boosted EfficientNet: Detection of Lymph Node Metastases in Breast Cancer Using Convolutional Neural Networks. Cancers 13(4):1–14. https://doi.org/10.3390/cancers13040661
86. What is Big-O complexity of classifying an image using CNN?, Cross Validated (n.d.) Available at: https://stats.stackexchange.com/questions/527142/what-is-big-o-complexity-of-classifying-an-image-using-cnn, Accessed on: 20-05-2022
87. What is the computational complexity of the forward pass of a convolutional neural network?, Artificial Intelligence (n.d.) Available at: https://ai.stackexchange.com/questions/22913/what-is-the-computational-complexity-of-the-forward-pass-of-a-convolutional-neur/22929#22929, Accessed on: 20-05-2022
88. Zhang, C, Bengio, S, Hardt, M, Recht, B, Vinyals, O (2017) Understanding Deep Learning Requires Rethinking Generalization, ICLR, pp.1–15, 2017 https://doi.org/10.48550/arXiv.1611.03530
89. Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 8697–8710 arXiv:1707.07012v4 [cs.CV]. https://doi.org/10.48550/arXiv.1707.07012