Landmark Recognition Model for Smart Tourism using Lightweight Deep Learning and Linear Discriminant Analysis

Mohd Norhisham Razali[1], Enurt Owens Nixon Tony[2], Ag Asri Ag Ibrahim[3], Rozita Hanapi[4], Zamhar Iswandono[5]
Faculty of Business and Management, Universiti Teknologi MARA Sarawak, Kota Samarahan, Malaysia[1, 4]
Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia[2, 3]
Higher Colleges of Technology, United Arab Emirates[5]

***Abstract*—Scene recognition algorithm is crucial for landmark recognition model development. Landmark recognition model is one of the main modules in the intelligent tour guide system architecture for the use of smart tourism industry. However, recognizing the tourist landmarks in the public places are challenging due to the common structure and the complexity of scene objects such as building, monuments and parks. Hence, this study proposes a super lightweight and robust landmark recognition model by using the combination of Convolutional Neural Network (CNN) and Linear Discriminant Analysis (LDA) approaches. The landmark recognition model was evaluated by using several pretrained CNN architectures for feature extraction. Then, several feature selections and machine learning algorithms were also evaluated to produce a super lightweight and robust landmark recognition model. The evaluations were performed on UMS landmark dataset and Scene-15 dataset. The results from the experiments have found that the Efficient Net (EFFNET) with CNN classifier are the best feature extraction and classifier. EFFNET-CNN achieved 100% and 94.26% classification accuracy on UMS-Scene and Scene-15 dataset respectively. Moreover, the feature dimensions created by EFFNet are more compact compared to the other features and even have significantly reduced for more than 90% by using Linear Discriminant Analysis (LDA) without jeopardizing classification performance but yet improved its performance.**

***Keywords*—Scene recognition; convolutional neural network; smart tourism; feature selections**

I. Introduction

Scene recognition is a crucial aspect for the development of many software applications such as in the area of intelligent robotics, autonomous driving and intelligent video surveillance. Moreover, scene recognition is the basis component in accomplishing the tasks for any object detection tasks [1]. The basic goal of scene recognition is to label all photos of scenes, whether they are outdoor or indoor, semantically and properly.

The magnificent scenery as well as the beautiful and historical landmarks of certain places has become one of the attraction factors for the tourist to come and visit these places. In this context, a software application that equipped with an intelligent landmark detection based on scene recognition algorithm can be developed to serve certain useful tasks. For instance, a tourist may get useful information and recommendation based on the detected landmark such as the nearby food attractions, and transportation and accommodation information. Besides, the application may assist the tourist agent while guiding the tourist visiting the attraction places. However, the scene recognition is a challenging task due to the difficulty to distinguish the common structure of the public scene objects such building, monuments, parks, beaches and so on [2]. Scene images also might be captured from different angles which triggered the high intra-class difference problems [3].

Deep learning and transfer learning based classification is the emergence approach in any machine learning tasks [4]–[6]. In scene recognition, the pretrained CNN models by using ResNet50 architecture were adopted [4], [5]. Although the classification accuracy obtained was good (92.17% and 94.4%), ResNet50 produced larger features dimensionality. Therefore, there are lot of studies in the other domain have various of Efficient Net (EFFNET) CNN architectures such as masked face recognition [7], smoke detection [8], chest X-ray scanning [9]–[11] and fake face video detection [12] due to its exceptional classification performance as well as to generate lightweight features.

The key contributions of this paper are the proposed super lightweight Landmark recognition model trained by using Convolutional Neural Network (CNN) to address the challenges of distinguishing the common public structure of landmark scenes. The features extracted by using the pretrained CNN model of EfficientNet (EFFNET) which produced the lightest features as compared to the other CNN models. Afterwards, Linear Discriminant Analysis (LDA) feature selection algorithm has been adopted that has significantly reduced the dimensionality of features without sacrificing classification performance at all and even have improved the classification performance. The recognition model training by using CNN was also very efficient as it required very minimal number epoch to complete and yield the best classification performance.

The remainder of the paper is organized in the following way: Section II provide the previous studies conducted in scene recognition. In Section III, the Methodology is described in more detail. Sections IV presents the experimental results. The conclusions and directions for the future studies are presented in Section V.

II. RELATED WORKS

Scene recognition is a subset of object recognition and can be treated as classification problem to serve certain purposes. It is a problem to describe the content or the objects that exist in the outdoor or indoor scene images. Scene recognition algorithms have adopted in many areas in computing field such as human computer interaction, robotics, smart surveillance system and autonomous driving [1]. Besides, scene recognition was also studied for tourism industry in assisting tourist or tourist guide to recognize the tourism attractive places or landmarks. There are Monulens [13], a real-time mobile-based landmark recognition, Smart Travelling [14] that used to recognize tourist attraction, nearby events, police stations and hospitals, Augmented Reality (AR) based landmark detection [15] and a system to distinguish large number of landmarks. All the aforementioned applications used the handcrafted features such as Histogram of Oriented Gradients (HOG), Scale Invariant Feature Transform (SIFT) and Bag of Features (BoF), and traditional machine learning approach such as Support Vector Machine (SVM). The recent works of scene recognition have shifted to deep learning-based approaches as tabulated in Table I.

TABLE I. RELATED WORKS OF SCENE RECOGNITION

Authors	Dataset	Techniques	Results
[16]	Places image	ImageNet-Linear SVM	Accuracy - 91.9%
[2]	Landmark database	DEep Local Features (DELF)	Specificity-0.99
[4]	Tourist Attraction Images	ResNet50	Accuracy - 92.17%
[5]	Scene images	ResNet-CNN	Accuracy - 94.4%

The study conducted in [16] established Places dataset to benchmark the performance of scene recognition algorithm which was denser in term of density and diversity of scene images in comparisons to the other scene recognition benchmark datasets such as SUN, Scene-15 and MIT Indoor67. The scene recognition algorithm trained by using Places dataset outperformed the accuracy performance of scene recognition algorithm trained by using ImageNet dataset for all scene recognition benchmark datasets. The evaluations were carried out by using CNN based features and linear SVM as classifier. The problem of high density and diversity of scene images as well as to determine whether the scene images contained landmark objects have been also addressed in [2] study. A metric learning-based approach was proposed in which the CNN is trained by curriculum learning technique and updated version of Center loss to overcome large variations of scene images. On the other hand, the existence of landmark objects in scene images determined by calculating distance between the image embedding vector and one or more centroids per class. Other than landmarks diversity issue, the scene recognition algorithm is also facing the high inter-class similarities where numerous landmarks have very similar building or architecture design. To overcome this problem, the CNN model based on ResNet50 was adopted in [4] to classify tourist attraction places in Jakarta, Indonesia such as Cathedral Church, Jakarta Old Town, Istiqlal Mosque and Maritime Museum. The ResNet based model also demonstrated exceptional performance in [5] via its proposed method namely Scene-RecNet to classify the aerial scene views such as airports, forests and rivers. The Scene-RecNet was more versatile and stable as the features are adjusted and modified in the convolutional and fully-connected layers that eventually improved the processing speed, small storage space and good recognition accuracy.

Table II shows the summary of previous studies that have adopted deep learning approaches, specifically transfer models.

TABLE II. RELATED WORKS OF DEEP LEARNING

Authors	Dataset	Techniques	Results
[17]	Land images	LeNet+Bagging based CNN	Recall-0.784
[7]	Face Mask	EFFNET based CNN	Accuracy-0.9972
[18]	Computer Tomography (CT) Images	Fusion of a moment invariant (MI) method+ ResNet150+VGG16	Accuracy-0.93
[8]	Smoke detection images	EFFNET based CNN	Accuracy-0.9818
[9]	Chest X-ray	DenseNet+EFFNetB0+Bi-LSTM	Accuracy-92.489%
[10]	Chest X-ray	EfficientNet-B2+CNN	Accuracy-96.33%
[12]	Fake Face Video	EFFNetB5+CNN	Accuracy-74.4%
[11]	Chest X-ray	EFFNetB0+CNN	Accuracy - 95.82%
[19]	TripAdvisor and Google	CNN	Accuracy-46.4%

The study conducted in [17] addressed the problem of land-use classification at the hilly and mountainous area by using ensemble learning approaches to improve the overall classification accuracy performance and classes number optimization to solve classification accuracy problem for coniferous forest. The bagging-based CNN using Bagging (Bootstrap AGGregatING) ensemble classifier is capable to overcome the problem of unstable procedures which means the great impact on classification due to minor differences of the data. Whereby the optimization of the classes' number was carried out by utilizing spectral clustering (SC) that divides data into subsets based on its similarity. The pre-trained LeNet CNN architecture have used for feature extraction. The pre-trained CNN architecture was also proposed in [18] for automatic screening of COVID-19. Specifically, two pre-trained CNN architecture ResNet50 and VGG16 were fused with the combination of Moment Invariant methods that improved the performance of previous COVID-19 classification models. It is also worth to note that many previous studies were adopted variant of EfficientNet (EFFNET) CNN architectures for extracting the features from the X-ray to detect lung related diseases. A variant of EFFNET namely EFFNETB0 with Bi-LSTM was proposed by [9] detect Covid-19 faster and with high accuracy low cost on chest X-ray images. Along with that, the features from EFFNETB0 were fused with DenseNet121 and LAB and CIE color space. The model training was performed by using Bi-LSTM classifier that yield the best classification accuracy as compared to the other ensemble classifiers. Similar techniques were also used in [11] to detect COVID-19 from lung X-ray. The other variant of EFFNET so called EFFNET B2 was found to be most effective as compared to the other variants in [10] to reduce the class imbalance problem for diagnosing pneumonia from chest X-RAY. The fine tuning on EFFNET architecture provides desirable impacts which reduce computational effort and the use of batteries. The evaluation of several EFFNET variants were also carried out in [12] to detect fake face video in social media website. Based on the evaluation, the optimal performance of detection is by using EFFNET B4 and B5 and the classification accuracy performance drops when using EFFNET B6 and B7. Next, The EFFNET with Linear SVM were used to address the issues images complexity to recognize the face mask wearing in [7] . In this study, the classification accuracy EFFNET has outperformed the other CNN models using DENSENET201, NASNETLARGE and INCEPTIONRESNETV2 with very light size of features. The lightness of features produced by EFFNET have been utilized by [8] through the proposed novel lightweight smoke detection for detecting fire in its early stages. A module for smoke region segmentation was also proposed in this study where the encoder-decoder approach with atrous separable convolutions were investigated.

According to the comprehensive survey conducted by [1], the top three performance recognition approaches fall under Patch Feature Encoding, Discriminative Region Detection and Hybrid Deep Models. Specifically, the CNN based feature extraction using ResNet-152, AlexNet and SE-ResNeXt-101 were recorded the significant performance on Scene-15, Sports-8, Indoor-67 and SUN-397.

Based on the discussions of the previous studies, it can be summarized that the pretrained CNN architecture is flexible and capable to provide robust recognition performance in various fields and domains. The CNN architecture is flexible as the layers and its parameters can be easily fine tuned to fit the requirement of data and optimum performance could be achieved. In particular, the EFFNET based CNN architecture has proven quite decent performance so far in terms of classification performance as well as to produce lightweight features. Therefore, the use EFFNET also might be extended in the domain of scene recognition to overcome the issue of high inter-class similarity in scene images.

III. Methodology

This section describes the methodology undertaken to carry out this research, as depicted in Fig. 2. The methodology consists of four parts which are data acquisition, feature extraction, feature selection and model training.

A. Experimental Setup

The experiment in this study was performed by using Python libraries based on Spyder 4.2.2 and PyCharm 2020.3.3 (Community Edition) software tools. Specifically, the feature extractions and classifications were performed by using Scikit-learn and Keras libraries.

B. Scene Recognition Model Training

The landmark recognition model training consists of four main steps which are data acquisition, feature extraction, feature selection and classification model training.

1) Data acquisition: The images for UMS Landmark Dataset were captured with a Nikon D7100 camera with a resolution of 6000 × 4000 pixels between 10.00 am. and 11.00 am. Fig. 1 shows the image samples of the popular landmarks in UMS [20]. This dataset has been made public and is available for download on the Kaggle website [21].

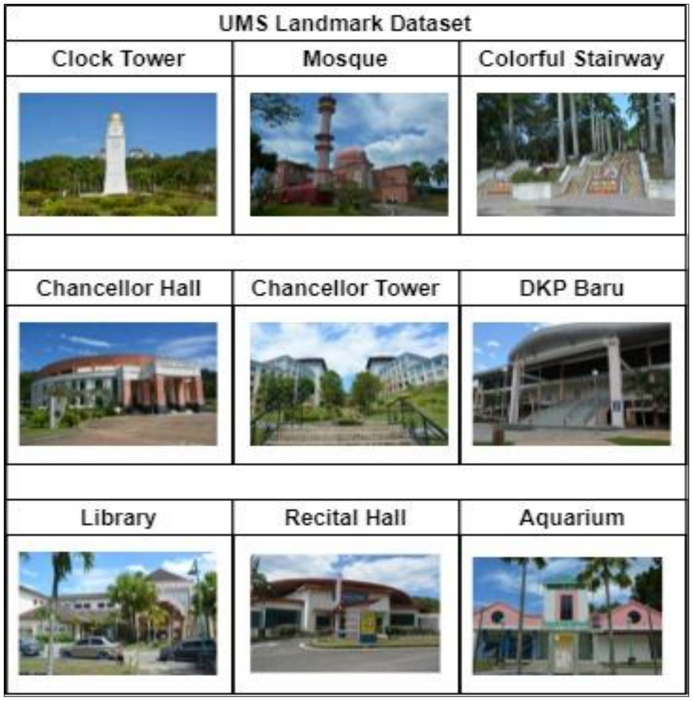


Fig. 1. Samples of UMS landmark dataset

Based on Fig. 1, there are nine categories of landmark consisting around 100 images with different camera angles. These landmarks are the popular tourist attractions for sightseeing and photography. Aside from this dataset, the public Scene-15 dataset [22] for scene recognition benchmarking was also evaluated in order to test the efficacy of the landmark recognition algorithm. This dataset contained 15 scene categories, comprising outdoor and indoor sceneries. There were 200 to 400 images in each category with an average resolution of 300 X 250 pixels.

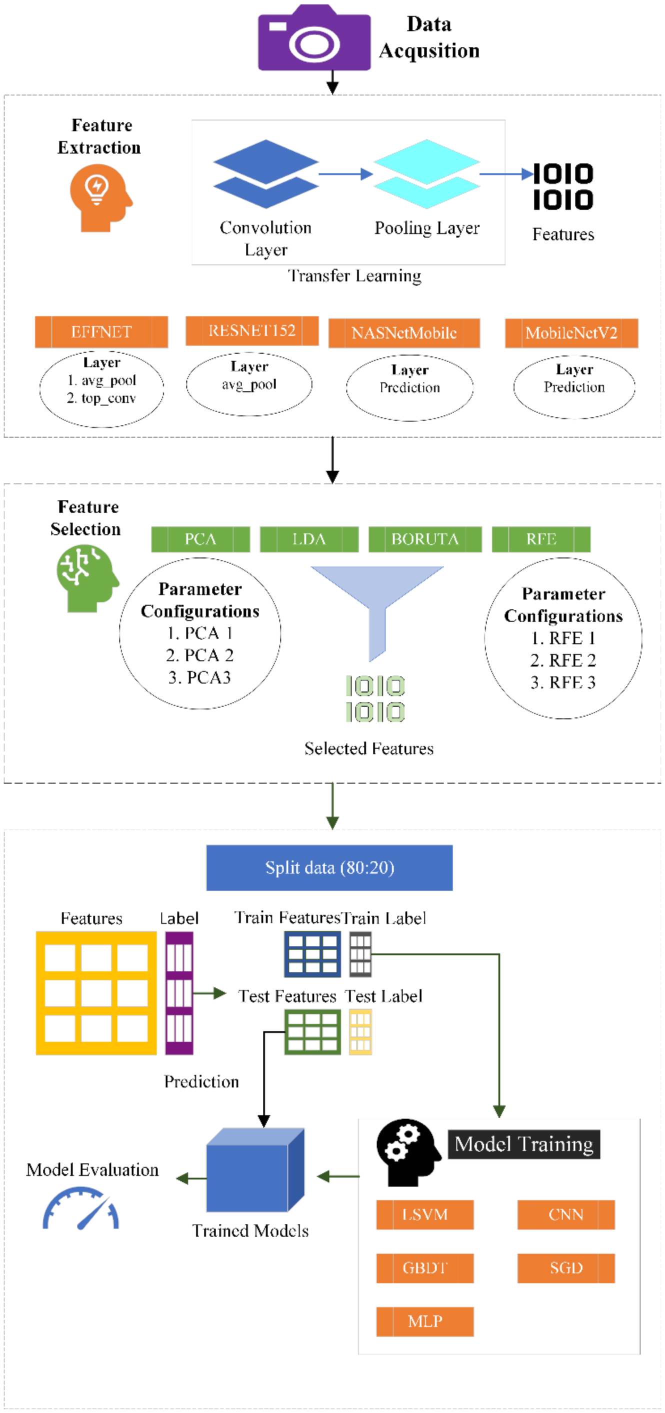


Fig. 2. Methodology

2) Feature extraction: Feature extraction is a process to transform the representation of the data into meaningful semantics for determining the category of the data in classification. In this work, the feature extraction was carried out by using transfer learning approach. The features of the images were extracted by re-using the model weights on the pre-trained Convolutional Neural Network (CNN) model. Transfer learning reduces the time it takes to train a neural network model and lead in decreasing generalization error. The extracted features of an image had created a vector of values that the model would use to characterize the image features. These characteristics were used in designing a new model.

In particular, four pre-trained CNN model were evaluated for feature extraction, which are Efficient Net (EFFNET) [23], RESNET152 [24], NASNetMobile [25] and MobileNetV2 [26]. EFFNET has been adopted and demonstrated to have an outstanding performance in recent studies such as in the Covid-19 detection based on chest X-Ray [9], [27], smoke detection [8], fake video detection [12], pneumonia classification [10], masked face detection [7] and food recognition [28]. Meanwhile, the RESNET152 was also reported to have a good performance for scene recognition [1].

Many previous studies have shown that the NASNeTMobile model performs well, such as the classification of rice diseases with an accuracy of 85.9% [29], ECG signal classification for cardiac examination [30] with an accuracy of 97.1 %, lung nodule classification from CT lung images with an accuracy of 88.28% [31] and skin lesion classification from dermoscopic images with an accuracy of 88.28% [32]. For on-device and embedded applications, the proposed MobileNetV2 has a low-latency, low-computation architecture. For instance, MobileNetV2 was used as an embedded food recognizer [33].

The pretrained CNN models were built with various layer types. In this work, two layer types of EFFNET layer were chosen to generate the feature matrices, namely *top_cov* and *avg_pool*. The model weights used in the EFFNET were ImageNet and both layers produced 62,720 and 1,280 feature dimensions. On the other hand, the *avg_pool* was the selected layer to generate 2048 features dimensions for RESNET152 model. Then, both NASNetMobile and MobileNetV2 produced 1000 feature dimensions.

The extracted features consist of one dimensional (1D) features matrix which will be fed into the traditional machine learning classifiers and the 1D CNN classifier (Conv1D). To work with 2D CNN classifier (Conv2D), the 1D features matrix was reshaped into 2D features matrix. The *top_cov* and *avg_pool* layers in EFFNET produced (16, 16, 5) and (16,16,245) output shape after being reshaped. Meanwhile, the *avg_pool* layer of RESNET152 produced (32, 32, 2) feature shape after being reshaped. Meanwhile the *prediction* layer of NASNeTMobile and MobileNetV2 generated a (2, 2, 250) feature shape. The feature shape represents the height, width and depth of the images which make the edge and colors of the spatial features to be detected.

3) Classification model: The extracted Conv1D or 1D features as described in (2) were fed to Linear Support Vector Machine (LSVM), CNN (1D), Gradient-Boosting Decision Tree (GBDT), Stochastic Gradient Descent (SGD) and Multilayer Perceptron (MLP). Linear kernel is applied and one-versus all (OVA) training strategy is used in LSVM. The parameters used for the classifiers during the experiment are shown in Tables III, IV, V and VI.

TABLE III. LSVM PARAMETERS

Parameters	Value	Description
Penalty	l2	Specifies the norm used in the penalization. The ‘l1’ leads to coef_ vectors that are sparse.
Loss	square_hinge	Specifies the loss function. ‘hinge’ is the standard SVM loss (used e.g. by the SVC class) while ‘squared_hinge’ is the square of the hinge loss.
Dual	1e-4	Tolerance for stopping criteria.
C	1.0	Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive.
Multi-class	ovr	Determines the multi-class strategy if y contains more than two classes. "ovr" trains n_classes one-vs-rest classifiers, while "crammer_singer" optimizes a joint objective over all classes

TABLE IV. GBDT PARAMETERS

Parameters	Value	Description
Loss	deviance	The loss function to be optimized. ‘deviance’ refers to deviance (= logistic regression) for classification with probabilistic outputs
learning_rate	0.1	Learning rate shrinks the contribution of each tree by learning_rate. There is a trade-off between learning_rate and n_estimators.
n_estimators	100	The number of boosting stages to perform. Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance.
subsample	1.0	The fraction of samples to be used for fitting the individual base learners. If smaller than 1.0 this results in Stochastic Gradient Boosting. subsample interacts with the parameter n_estimators. Choosing subsample < 1.0 leads to a reduction of variance and an increase in bias.
criterion	friedman_mse	The function to measure the quality of a split

TABLE V. SGD PARAMETERS

Parameters	Value	Description
Loss	hinge	Defaults to ‘hinge’, which gives a linear SVM
penalty	l2	Defaults to ‘l2’ which is the standard regularizer for linear SVM models
alpha	0.0001	Constant that multiplies the regularization term
fit_intercept	True	Whether the intercept should be estimated or not. If False, the data is assumed to be already centered.
max_iter	1000	The maximum number of passes over the training data (aka epochs). It only impacts the behavior in the fit method, and not the partial_fit method.

TABLE VI. MLP PARAMETERS

Parameters	Value	Description
hidden_layer_sizes	(100,)	The ith element represents the number of neurons in the ith hidden layer.
activation	relu	Activation function for the hidden layer.
solver	adam	The solver for weight optimization.
alpha	0.0001	0.0001
batch_size	auto	Size of minibatches for stochastic optimizers
learning_rate	constant	Learning rate schedule for weight updates.

On the other hand, the Conv2D training features produced by EFFNET were fed into 2D Convolutional Neural Network classifier which is a fully connected layer. Table VII shows all the layers, the output shapes and the total parameters for EFFNET (*avg_pool*), EFFNET (*top_conv*) and RESNET152.

TABLE VII. 2D CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE

Layer (type)	**Output Shape**			**Parameters**		
	EFF NET (AvgPool)	***EFF NET (TopConv)***	***RES NET152***	***EFFNET (AvgPool)***	***EFFNET (TopConv)***	***RES NET152***
conv2d (Conv2D)	(None, 16, 16, 32)	(None, 16, 16, 32)	(None, 32, 32, 32)	1472	70592	608
dropout (Dropout)	(None, 16, 16, 32)	(None, 16, 16, 32)	(None, 32, 32, 32)	0	0	0
conv2d_1 (Conv2D)	(None, 14, 14, 32)	(None, 14, 14, 32)	(None, 30, 30, 32)	9248	9248	9248
max_pooling 2d (MaxPooling 2D)	(None, 7, 7, 32)	(None, 7, 7, 32)	(None, 15, 15, 32)	0	0	0
flatten (Flatten)	(None, 1568)	(None, 1568)	(None, 7200)	0	0	0
dense (Dense)	(None, 512)	(None, 512)	(None, 512)	803328	803328	3686912
dropout_1 (Dropout)	(None, 512)	(None, 512)	(None, 512)	0	0	0
dense_1 (Dense)	(None, 12)	(None, 12)	(None, 12)	6156	6156	6156
Total params	820,204	883,168	3,702,924			
Trainable params	820,204	883,168	3,702,924			
Non-trainable params	0	0	0			

CNN possesses convolution layer that has several filters to perform the convolution operation, which are RELU, pooling layer, and fully connected layer. The RELU layer produces the rectified feature map by performing the operation on the elements. The rectified feature map next feeds into a pooling layer. Pooling is a down-sampling operation that reduces the dimensions of the feature map. The rectified feature map is fed into a pooling layer. Pooling is a down-sampling operation that decreases the feature map's dimensionality. By flattening the two-dimensional arrays from the pooled feature map, the pooling layer turns them into a single, long, continuous, linear vector. When the flattened matrix from the pooling layer is given as an input, a fully connected layer forms classifies the images.

The dataset will undergo training and testing phase in creating the classification model. In CNN, the epoch refers to the number of times the model trains all datasets. Whereby, batch size is a small amount of data used for training. A suitable number of epochs needs to be adjusted until a small gap between test and training error can be observed. When the appropriate number of epochs is not chosen, underfitting and overfitting problems occur.

The learning rate determines how frequently the weight in the optimization method is updated. Fixed learning rate is used and the Adam is chosen as optimizer.

Dropout is a better regularization strategy for deep neural networks to avoid overfitting. The method essentially removes units from a neural network based on the probability desired. A default value of 0.5 is set in this experiment. Loss function measure the successfulness of classification and in this experiment by defining the distance between two data points. In this experiment, the categorical cross-entropy loss function was used.

4) Feature selection: Feature selection plays important roles to improve the performance of recognition model by reducing the features dimensionality and transforming the feature into meaningful features [34], [35]. The meaningful features are characterized by the features that are more salient, less overfit and reduced the training execution time which eventually improve the accuracy performance [36]. In this work, Principal Component Analysis (PCA) [37], Linear Discriminant Analysis (LDA) [38], Boruta [39] and Recursive Feature Elimination (RFE)[40] were evaluated.

Table VIII shows the number of features selected after performing the feature selection algorithms. Unlike PCA, LDA and RFE, Boruta provided automatic mechanism in determining the number of features. Therefore, manual parameter configurations to determine the number of features selected were not required. Meanwhile, the number of features selected for LDA need to be set to less or equal to the total class in the dataset. For PCA and RFE, experiments were conducted to test three configurations with different percentages of features selected, which are 70%, 40% and 10%.

TABLE VIII. NUMBER OF FEATURES SELECTED

Feature Selection	Configurations	Features	UMS Dataset	Scene-15 Dataset
PCA	PCA1 (70%)	EFFNET	896	
		RESNET152	819	
		NASNETMobile	700	
		MobileNetV2	700	
	PCA2 (40%)	EFFNET	512	
		RESNET152	205	
		NASNETMobile	400	
		MobileNetV2	400	
	PCA3 (10%)	EFFNET	128	
		NASNETMobile	100	
		MobileNetV2	100	
LDA		EFFNET	8	14
		RESNET152		
		NASNETMobile		
		MobileNetV2		
BORUTA		EFFNET	749	372
		RESNET152	479	149
		NASNETMobile	677	61
		MobileNetV2	777	158
RFE	RFE1 (70%)	EFFNET	896	
		RESNET152	1434	
		NASNETMobile	700	
		MobileNetV2	700	
	RFE2 (40%)	EFFNET	512	
		RESNET152	819	
		NASNETMobile	400	
		MobileNetV2	400	
	RFE2 (10%)	EFFNET	128	
		RESNET152	205	
		NASNETMobile	100	
		MobileNetV2	100	

5) *Classification model performance metrics:* The model's overall performance on the testing set was measured using the accuracy metric as the performance metric. Assume that CM is a n by n confusion matrix, with n equaling the total number of various scene categories. Next, the actual category is indicated by the row of CM, while the anticipated category is indicated by the column of CM. Then, let C (i,j) denote the value of the CM cell in index row I and column j, with i,j=1,2,...,n. The following equation defined the accuracy metrics:

$$accuracy = \frac{\sum_{i,j=1}^{n} c_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{n} c_{i,j}} \quad (1)$$

IV. Findings

This section presents the analysis from the experiment results comprising feature extraction, classification and feature selection performance. The first part of this section presents the discussion of classification performance evaluation, the second part discusses about the feature dimensions size, shape and the number of epoch used in CNN training, followed by the performance analysis for feature selection.

A. Classification Performance

Table IX shows the recognition accuracy of feature extraction based on EFFNET, RESNET152, NASNetMobile and MobileNetV2 and classification by using Linear SVM (LSVM), CNN (2D), CNN (1D), Gradient-Boosting Decision Tree (GBDT), Stochastic Gradient Descent (SGD) and Multilayer Perceptron (MLP) on UMS-Scene and Scene-15 dataset.

TABLE IX. Classification Accuracy Comparisons between UMS Landmark and Scene-15 Dataset

Feature Extraction	Layer Name	Classification	UMS-Scene	Scene-15
EFFNET 1	*avg_pool*	LSVM	**1.00**	0.94
		CNN (2D)	**1.00**	0.85
		CNN (1D)	**1.00**	**0.94**
		GBDT	**1.00**	0.68
		SGD	**1.00**	0.68
		MLP	0.44	0.43
EFFNET 2	*top_conv*	LSVM	0.94	**0.94**
		CNN (2D)	0.95	0.91
		CNN (1D)	**1.00**	0.92
		GBDT	**1.00**	0.66
		SGD	**1.00**	0.92
		MLP	0.12	0.40
RESNet152	*avg_pool*	LSVM	**1.00**	**0.62**
		CNN (2D)	0.85	0.58
		CNN (1D)	**1.00**	0.62
		GBDT	**1.00**	0.41
		SGD	0.95	0.37
		MLP	0.12	0.23
NASNetMobile	*prediction*	LSVM	0.77	0.68
		CNN (2D)	0.13	0.38
		CNN (1D)	**1.00**	**0.74**
		GBDT	0.99	0.55
		SGD	0.82	0.58
		MLP	0.33	0.39
MobileNetV2	*prediction*	LSVM	0.85	0.69
		CNN (2D)	0.13	0.36
		CNN (1D)	**1.00**	**0.82**
		GBDT	**1.00**	0.07
		SGD	0.77	0.68
		MLP	0.56	0.34

In comparison to the Scene-15 dataset, most of the algorithms performed well on the UMS landmark dataset, as shown in Table IX. As the UMS landmark dataset had a higher image resolution, the quality of the collected images was more likely to have influenced the result. The bar charts in Fig. 3, Fig. 4, Fig. 5, and Fig. 6 show how the features and classifiers employed in the UMS landmark and Scene-15 datasets compare in terms of performance. The classification accuracy of various features on various classifiers is shown in Fig.3. EFFNET with avg_pool layer is the best feature due to its perfect achievement on all classifiers except MLP. To demonstrate its efficacy, Fig. 4 shows the classification accuracy of various classifiers on various features. Except for NASNetMobile, CNN 1D and GBDT had been found to be resilient to a variety of features, including the ability to attain 100% classification accuracy on all features. In contrary, CNN 2D performed poorly with NASNetMobile and MobileNetV2. This was most likely because the 2D shape features generated by the CNN 2D classifier were incompatible.

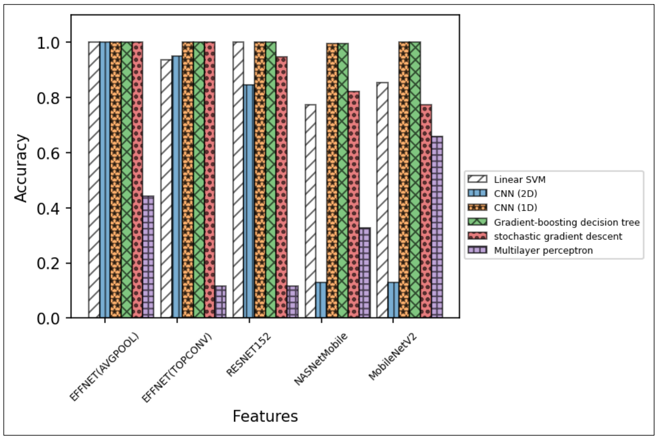


Fig. 3. Performance of features on UMS landmark dataset

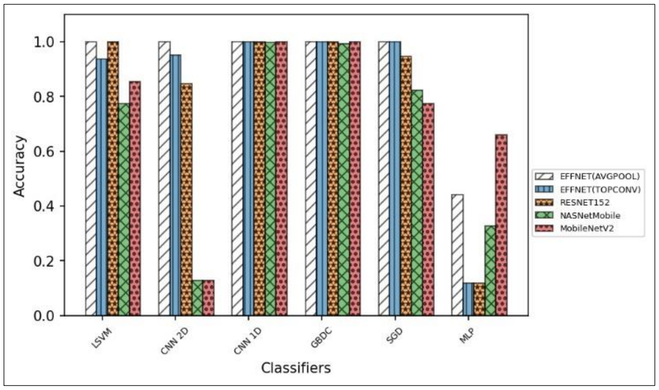


Fig. 4. Performance of classifiers on UMS-landmark dataset

EFFNET based features performed well across many classifiers in the Scene-15 dataset, apart from GBDC and MLP, as shown in Fig. 5. RESNet152, NASNetMobile, and MobileNetV2, on the other hand, produced less discriminative features. Fig. 6 shows that LSVM and CNN 1D perform consistently across all features and worked exceptionally well with EFFNET features. GBDC and MLP, on the other hand, only achieved 67.61% and 43.39% accuracy, respectively. Moreover, the CNN 2D and SGD only worked well with EFFNET features. Overall, the best classification accuracy on the Scene-15 dataset was 94.26% using CNN 1D classifier and EFFNET (AVGPOOL) features. Based on the study conducted in [1], the RESNet152 indeed yielded the best performance on Scene-15, Sports-8, Indoor-67 and SUN-397. However, based on the result of the experiment in this paper revealed that the EFFNET have better performance on Scene-15 dataset. Next, the confusion matrix of classification accuracy is illustrated in Fig. 7.

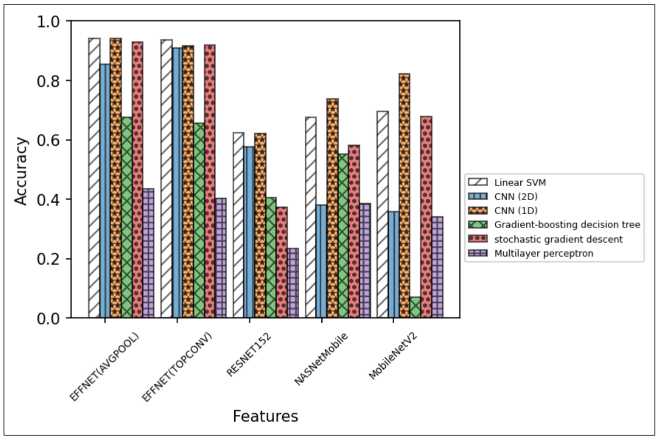


Fig. 5. Performance on features on scene-15 dataset

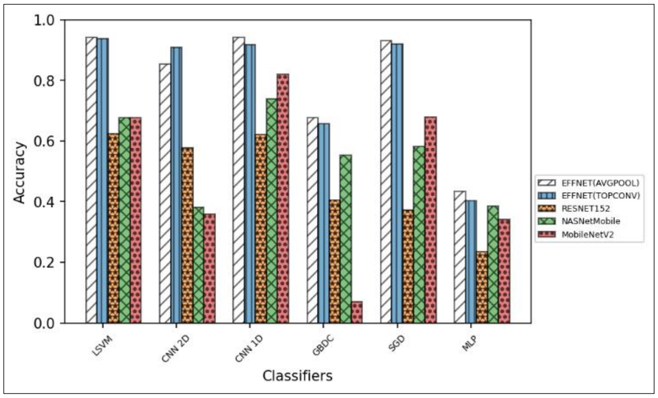
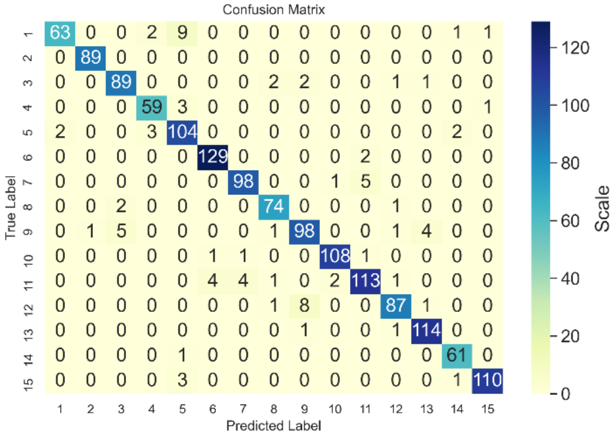


Fig. 6. Performance of classifiers on scene-15 dataset



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	63	0	0	2	9	0	0	0	0	0	0	0	0	1	1
2	0	89	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	89	0	0	0	0	2	2	0	0	1	1	0	0
4	0	0	0	59	3	0	0	0	0	0	0	0	0	0	1
5	2	0	0	3	104	0	0	0	0	0	0	0	0	2	0
6	0	0	0	0	0	129	0	0	0	0	2	0	0	0	0
7	0	0	0	0	0	0	98	0	0	1	5	0	0	0	0
8	0	0	2	0	0	0	0	74	0	0	0	1	0	0	0
9	0	1	5	0	0	0	0	1	98	0	0	1	4	0	0
10	0	0	0	0	0	1	1	0	0	108	1	0	0	0	0
11	0	0	0	0	0	4	4	1	0	2	113	1	0	0	0
12	0	0	0	0	0	0	0	1	8	0	0	87	1	0	0
13	0	0	0	0	0	0	0	0	1	0	0	1	114	0	0
14	0	0	0	0	1	0	0	0	0	0	0	0	0	61	0
15	0	0	0	0	3	0	0	0	0	0	0	0	0	1	110

Fig. 7. Confusion matrix of classification using CNN 1D-EFFNET (AVGPOOL) on Scene-15 dataset

As plotted in Fig. 7, there are few scene images had been miscategorized. For instance, category 1 (office) was classified as category 5 (store), category 7 (tall building) was classified as category 11 (coast), category 9 (street) was classified as category 3 (living room), category 13 (mountain) was classified as category 9 (open country), and category 12 (open country) could be classified as category 9 (open country) (street). This shows that the high inter-class similarity classification problem still exists due to the appearance diversity of scene photos.

Table X and Table XI shows the precision, recall, F1-score and sup. (support) performance of the algorithms on UMS landmark dataset and Scene-15 dataset. Precision is the capability of a classifier to avoid classifying a negative instance as positive. It is described for each class as the proportion of true positives to the total of true positives and false positives. Recall is the capacity of a classifier to find all instances that are positive. It is described as the ratio of true positives to the total of true positives and false negatives for each class. A weighted harmonic mean of recall and precision is used to get the F1 score, with the best result being 1.0 and the lowest being 0.0. Due to the inclusion of precision and recall in their computation, F1 scores are lower than accuracy measurements. Support is the number of instances of the class that occur in the particular dataset. The requirement for stratified sampling or rebalancing may be indicated by unbalanced support in the training data, which may point to structural flaws in the classifier's reported scores. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing.

TABLE X. CLASSIFICATION PERFORMANCE ON UMS LANDMARK DATASET

Feature Extraction	Classifier	Prec.	Rec.	F1-Score	Sup.
EFFNet 1	LSVM	1	1	1	309
	CNN (2D)	0.99	0.99	0.99	281
	CNN (1D)	1	1	1	310
	GBDT	1	1	1	281
	SGD	1	1	1	281
	MLP	0.35	0.44	0.36	281
EFFNet 1	LSVM	1	1	1	309
	CNN (2D)	0.96	0.95	0.95	281
	CNN (1D)	1	1	1	310
	GBDT	1	1	1	282
	SGD	1	1	1	282
	MLP	0.01	0.12	0.02	282
RESNet152	LSVM	1	1	1	309
	CNN (2D)	0.86	0.85	0.84	281
	CNN (1D)	1	1	1	310
	GBDT	1	1	1	282
	SGD	0.95	0.95	0.94	282
	MLP	0.01	0.12	0.02	282
NASNetMobile	LSVM	0.82	0.77	0.74	282
	CNN (2D)	0.02	0.13	0.03	281
	CNN (1D)	1	1	1	310
	GBDT	0.99	0.99	0.99	282
	SGD	0.88	0.82	0.79	282
	MLP	0.34	0.33	0.27	282
MobileNetV2	LSVM	0.87	0.85	0.86	282
	CNN (2D)	0.02	0.13	0.03	281
	CNN (1D)	1	1	1	310
	GBDT	1	1	1	282
	SGD	0.88	0.77	0.75	282
	MLP	0.56	0.66	0.59	282

TABLE XI. CLASSIFICATION PERFORMANCE ON SCENE-15 DATASET

Feature Extraction	Classifier	Prec.	Rec.	F1-Score	Sup.
EFFNet 1	LSVM	0.94	0.94	0.94	1480
	CNN (2D)	0.83	0.83	0.83	1167
	CNN (1D)	0.94	0.94	0.94	1481
	GBDT	0.69	0.68	0.67	1346
	SGD	0.69	0.68	0.67	1346
	MLP	0.34	0.43	0.36	1346
EFFNet 2	LSVM	0.94	0.94	0.94	1480
	CNN (2D)	0.91	0.91	0.91	1167
	CNN (1D)	0.92	0.92	0.92	1481
	GBDT	0.68	0.66	0.6	1480
	SGD	0.92	0.91	0.92	1480
	MLP	0.43	0.4	0.5	1480
RESNet152	LSVM	0.63	0.62	0.63	1480
	CNN (2D)	0.56	0.55	0.54	1167
	CNN (1D)	0.63	0.62	0.62	1481
	GBDT	0.42	0.41	0.4	1346
	SGD	0.52	0.37	0.31	1346
	MLP	0.14	0.23	0.17	1346
NASNetMobile	LSVM	0.69	0.68	0.66	1346
	CNN (2D)	0.29	0.38	0.31	1350
	CNN (1D)	0.74	0.74	0.74	1481
	GBDT	0.52	0.55	0.5	1346
	SGD	0.67	0.58	0.57	1346
	MLP	0.25	0.39	0.29	1346
MobileNetV2	LSVM	0.7	0.69	0.69	1346
	CNN (2D)	0.28	0.36	0.3	1350
	CNN (1D)	0.82	0.82	0.82	1481
	GBDT	0.04	0.07	0.05	1346
	SGD	0.72	0.68	0.67	1346
	MLP	0.26	0.34	0.28	1346

B. Features Shape and Number of Epoch

The extracted features were reshaped into 1D and 2D representations, as can referred in Table XIII. The 1D feature shape was being fed to LSVM, CNN 1D, GBDT, SGD, and MLP, whereby the 2D feature shape was being fed to CNN 2D classifier. For both datasets, Table XII and Table XIII show the features' form as well as the best number of epoch for training the CNN. As seen in Table XII, the EFFNET generated the largest 1D features (62720) by using the average pool layer. NASNetMobile and MobileNetV2, on the other hand, generated the smallest number of features (1000). The best classification accuracy can be obtained by using only 30 epochs via CNN 1D for all the features. Whereby, the number of epochs was higher for training the CNN 2D are except for MobileNetV2.

TABLE XII. FEATURES' DIMENSION SIZE AND EPOCH FOR UMS LANDMARK DATASET

Feature Extraction	Layer Name	Features Shape (1D)	Features Shape (2D)	No.Epoch (CNN 1D)	No.Epoch (CNN 2D)
EFFNet	*avg_pool*	(1, 1280)	(16,16,5)	30	120
	top_conv	(1, 62720)	(16,16,245)	30	120
RESNet152	*avg_pool*	(1,2048)	(32,32,2)	30	150
NASNetMobile	*Prediction*	(1,1000)	(2,2,250)	30	60
MobileNetV2	*Prediction*	(1,1000)	(2,2,250)	30	30

TABLE XIII. FEATURES DIMENSIONS SIZE AND EPOCH DOR SCENE-15 DATASET

Feature Extraction	Layer Name	Features Length (1D)	Features Length (2D)	No.Epoch (CNN 1D)	No.Epoch (CNN 2D)
EFFNET	*avg_pool*	(1, 1280)	(16,16,5)	120	150
	top_conv	(1, 62720)	(16,16,245)	30	150
RESNET 152	*avg_pool*	(1,2048)	(32,32,2)	60	150
NASNetMobile	*Prediction*	(1,1000)	(2,2,250)	60	150
MobileNetV2	*Prediction*	(1,1000)	(2,2,250)	30	150

Based on Table XIII, the number of epoch required for training the CNN classifiers for Scene-15 dataset was larger than UMS landmark dataset. It was found that the CNN 2D required up to 150 epochs for CNN training.

Fig. 8, Fig. 9, Fig. 10, and Fig. 11 present the graph of model accuracy and model loss over number of epochs for EFFNET and MobilenetV2 by using CNN 2D and CNN 1D classifiers. To determine the appropriate number of epochs for each CNN architecture, the evaluation was made on 30, 60, 90, 120, and 150 epochs. By using 120 number of epoch, the EFFNET with *avg_pool* layer managed to obtain the best classification performance with very minimal gap between training and test model lost, as can be seen in Fig. 9. On the other hand, a slightly larger gap size can be observed between training and testing in model loss in EFFNET using *top_conv* layer with stagnant performance in model accuracy despite of larger number of epochs being used as shown in Fig. 11.

In summary, the feature extraction by using EFFNET by using *avg_pool* and *top_conv* layers with both CNN and SVM classifiers can be considered as the best option in this context and with their own merits. For instance, the EFFNET with *avg_pool* layer produced a light feature size which definitely use less computational effort for storage and classification. Meanwhile, the EFFNET with *top_conv* layer, even though it produced a larger size of features, but required a very minimum number of epochs to run the CNN classifier with a high classification accuracy. Thus, the trained model, by using EFFNET-*avg_pool* with CNN 1D classifier could be deployed in the development of Landmark Recognition System.

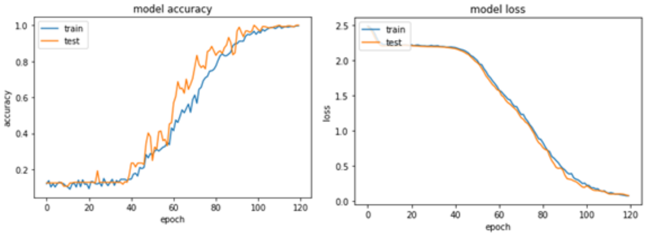


Fig. 8. EFFNET (AVGPOOL)- CNN 2D on UMS scene dataset

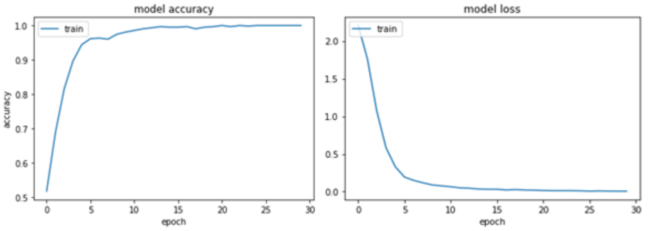


Fig. 9. MobileNetV2 – CNN 1D on UMS scene dataset

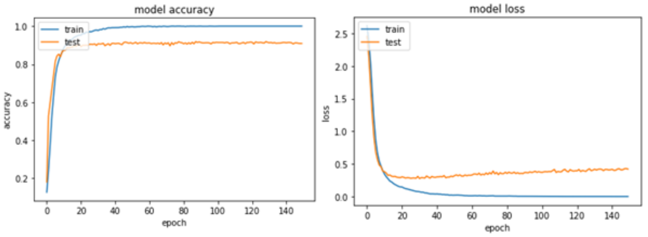


Fig. 10. EFFNET(TOPCONV)- CNN 2D on scene-15 dataset

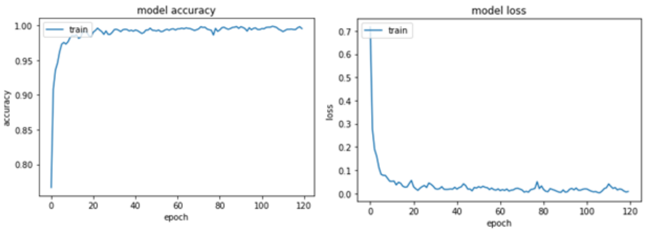


Fig. 11. EFFNET(AVGPOOL)- CNN 1D on scene 15 dataset

C. Effect of Feature Selection

The performance of feature selection methods such as PCA, LDA, Boruta, and RFE on the UMS and Scene-15 datasets, is discussed in this section. The feature selections were applied to EFFNET, RESNET152, NASNETMobile, and MobileNetV2 features, in particular.

1) UMS dataset: The three PCA variations, as shown in Table XIV, mirrored the varying proportions of features selected, as seen in Table IX. As shown in Table XIV, the baseline referred to the findings achieved in the prior trial without any treatment employing feature selection.

TABLE XIV. EFFECT OF PCA ON ACCURACY FOR UMS DATASET

Feature Extraction	Classificatio n	Baselin e	PCA 1	PCA 2	PCA 3	Spark -line
EFFNET	LSVM	1.00	1.00	1.00	**1.00**	—
	CNN (1D)	1.00	1.00	1.00	**1.00**	—
	GBDT	1.00	0.99	0.99	0.99	╲_
	SGD	1.00	1.00	1.00	**1.00**	—
	MLP	0.44	0.68	0.59	0.57	/‾
RESNET 152	LSVM	1.00	1.00	1.00	1.00	—
	CNN (1D)	1.00	1.00	1.00	1.00	—
	GBDT	1.00	0.98	0.96	0.95	╲
	SGD	0.95	1.00	1.00	1.00	/‾
	MLP	0.12	0.66	0.64	0.64	/‾
NASNet Mobile	LSVM	0.77	0.77	0.77	0.77	—
	CNN (1D)	1.00	1.00	1.00	**1.00**	—
	GBDT	0.99	0.91	0.92	0.98	╲_
	SGD	0.82	0.88	0.80	0.91	/\
	MLP	0.33	0.12	0.12	0.61	╲_
MobileNetV 2	LSVM	0.85	0.85	0.85	0.85	—
	CNN (1D)	1.00	1.00	1.00	**1.00**	—
	GBDT	1.00	0.88	0.87	0.89	╲_
	SGD	0.77	0.82	0.77	0.85	/\
	MLP	0.56	0.72	0.12	0.12	‾╲

Based on the overall result in Table XIV, the treatment of PCA had a positive effect on majority of the features as it has retained accuracy performance and even more, slight improvement on the accuracy can be observed on all the features especially the MLP classifiers. In a flipside, the accuracy performance using GBDT has slightly affected regardless any features used.

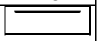
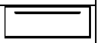
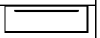
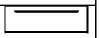
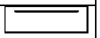
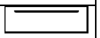
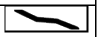
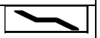
Table XV shows the classification performance after the LDA and Boruta were performed on all the features. The LDA had a positive effect on the accuracy performance for almost all the features except the classification using MLP. Despite pruning more that 90% of features by using LDA, the accuracy performance improvement can be observed on RESNET152, NASNETMobile and MobileNetV2 along sustaining the best accuracy performance on EFFNET. On the other hand, the BORUTA only demonstrated positive effect on EFFNET and RESNET152. The other highlight was the classification using MLP on EFFNET features has dramatically improved the accuracy performance from 0.44 to 0.83.

TABLE XV. EFFECTS OF LDA AND BORUTA ON ACCURACY FOR UMS DATASET

Feature Extraction	Classificatio n	Baselin e	LD A	BORUT A	Trendline
EFFNET	LSVM	1.00	**1.00**	1.00	■ ■ ■
	CNN (1D)	1.00	**1.00**	1.00	■ ■ ■
	GBDT	1.00	**1.00**	1.00	■ ■ ■
	SGD	1.00	**1.00**	1.00	■ ■ ■
	MLP	0.44	0.19	0.83	▬ _ ■
RESNET 152	LSVM	1.00	**1.00**	1.00	■ ■ ■
	CNN (1D)	1.00	**1.00**	1.00	■ ■ ■
	GBDT	1.00	**1.00**	1.00	■ ■ ■
	SGD	0.95	**1.00**	0.99	_ ■ ▬
	MLP	0.12	0.12	0.12	■ ■ ■
NASNet Mobile	LSVM	0.77	**1.00**	0.77	_ ■ _
	CNN (1D)	1.00	**1.00**	1.00	■ ■ ■
	GBDT	0.99	**1.00**	0.99	_ ■ _
	SGD	0.82	**1.00**	0.93	_ ■ ▬
	MLP	0.33	0.12	0.12	■ _ _
MobileNetV 2	LSVM	0.85	**1.00**	0.85	_ ■ _
	CNN (1D)	1.00	**1.00**	1.00	■ ■ ■
	GBDT	1.00	**1.00**	1.00	■ ■ ■
	SGD	0.77	**1.00**	0.77	_ ■ _
	MLP	0.56	0.19	0.12	■ ▬ _

Table XVI presents the analysis of feature selection performance using RFE.

TABLE XVI. EFFECTS OF RFE ON ACCURACY FOR UMS DATASET

Feature Extraction	Classification	Baseline	RFE 1	RFE 2	RFE 3	Spark-line
EFFNet	LSVM	1.00	1.00	1.00	**1.00**	
	CNN (1D)	1.00	1.00	1.00	**1.00**	
	GBDT	1.00	1.00	1.00	**1.00**	
	SGD	1.00	1.00	1.00	**1.00**	
	MLP	0.44	0.46	0.48	0.55	
RESNet152	LSVM	1.00	1.00	1.00	1.00	
	CNN (1D)	1.00	1.00	1.00	1.00	
	GBDT	1.00	1.00	1.00	0.96	
	SGD	0.95	0.93	1.00	1.00	
	MLP	0.12	0.39	0.12	0.64	
NASNet Mobile	LSVM	0.77	0.14	0.16	0.20	
	CNN (1D)	1.00	0.82	0.34	0.12	
	GBDT	0.99	0.96	0.98	0.99	
	SGD	0.82	0.44	0.39	0.10	
	MLP	0.33	0.12	0.12	0.12	
MobileNet V2	LSVM	0.85	0.13	0.23	0.22	
	CNN (1D)	1.00	1.00	0.77	0.17	
	GBDT	1.00	0.99	0.99	0.98	
	SGD	0.77	0.13	0.13	0.12	
	MLP	0.56	0.12	0.12	0.12	

RFE worked well on EFFNET and RESNET152, as shown in Table XVII. Moreover, the performance of MLP on RESNET152 had substantially improved from 0.12 to 0.64. On the other hand, RFE absolutely failed to perform on NASNetMobile and MobileNetV2, resulting in a significant fall in the accuracy of all classifiers used.

Next, the detailed analysis of feature selection performance on each feature and machine learning classifier are shown in Fig. 12, Fig. 13, Fig.14 and Fig.15.

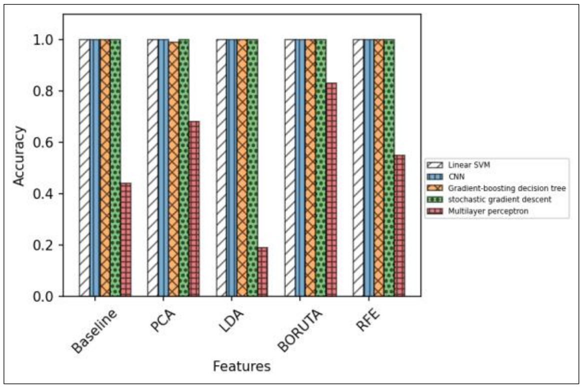


Fig. 12. Effects of feature selection on machine learning classifiers for EFFNET (UMS dataset)

As shown in Fig. 12, except for MLP, all classifiers in EFFNET performed remarkably well on all feature selections. Whereby, the EFFNET features would be more compatible with MLP if PCA and BORUTA is applied as the accuracies were increased by 55% and 89% respectively. On RESNET152 with EFFNET, a similar pattern of feature selection performance can be observed, as shown in Fig. 13. In fact, regardless of which feature selection is employed, the accuracy of SGD can be improved. When PCA and RFE were used with MLP, a positive effect on accuracy was noticed.

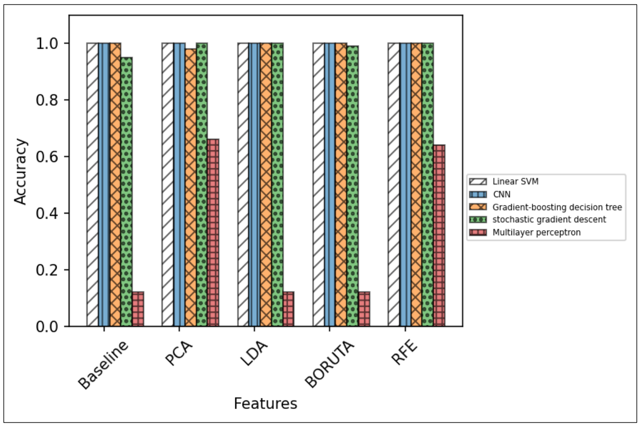


Fig. 13. Effects of feature selection on machine learning classifiers for RESNET152 (UMS dataset)

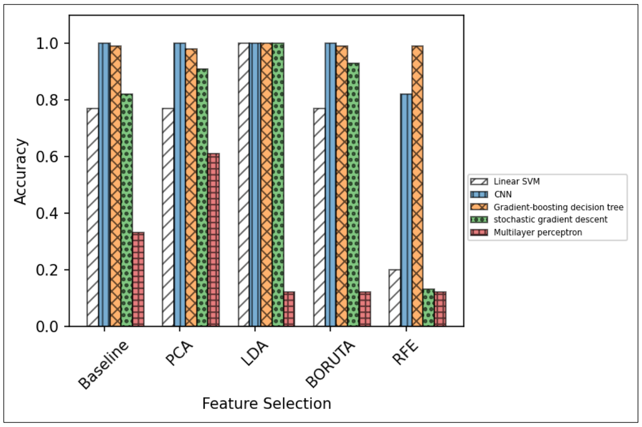


Fig. 14. Effects of feature selection on machine learning classifiers for NASNETMOBILE (UMS dataset)

According to the graph in Fig. 14, GBDT's performance appeared to be consistent across all feature selections, but the performance of the other classifiers dropped when RFE was applied. The best performance of LSVM and SGD could be seen when LDA was used. On MobileNetV2, CNN performed very well with all the feature selections and GBDT was slightly incompatible with PCA. Similar with NASNetMobile, LDA had also improved the accuracy of LSVM and SGD.

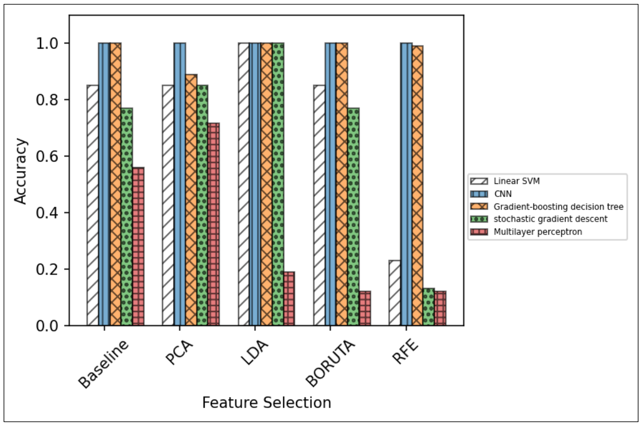


Fig. 15. Effects of feature selection on machine learning classifiers for MOBILENETV2 (UMS dataset)

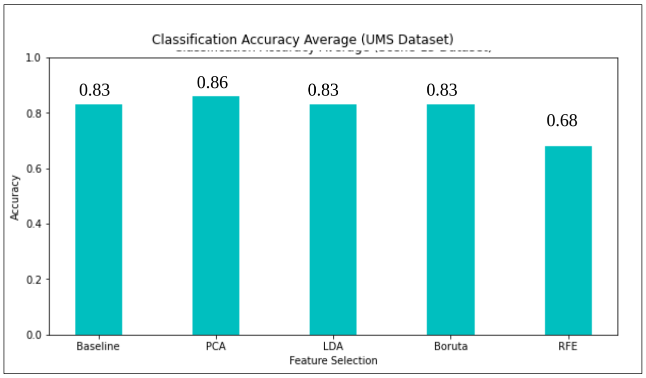
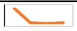


Fig. 16. The average performance comparisons of feature selection for UMS dataset

The summary of feature selection performance across features and classifiers for the UMS dataset is shown in Fig. 16. The PCA was found to be the most robust feature selection method since its performance was consistent across various features and classifiers. However, when accuracy and feature size were taken into account, LDA's performance was the most significant. Meanwhile, if execution time was not a major concern and automatic feature selection is one of the criteria for selecting features, the BORUTA could be considered. Aside from that, the results of Tables XII, XIII, and XIV implied that EFFNET is the best and stable features. The best classifiers were GBDT and CNN, which consistently excelled across a variety of feature selections.

2) Scene-15 dataset: Table XVII shows the performance analysis of PCA on Scene-15 dataset.

TABLE XVII. EFFECTS OF PCA ON ACCURACY FOR SCENE-15 DATASET

Feature Extraction	Classification	Baseline	PCA 1	PCA 2	PCA 3	Spark line
EFFNET	LSVM	0.94	0.94	0.93	0.91	
	CNN (1D)	0.94	0.93	0.92	0.93	
	GBDT	0.68	0.06	0.01	0.05	
	SGD	0.68	0.93	0.93	0.93	
	MLP	0.43	0.56	0.32	0.33	
RESNET152	LSVM	0.62	0.61	0.59	0.55	
	CNN (1D)	0.62	0.66	0.66	0.66	
	GBDT	0.41	0.33	0.44	0.49	
	SGD	0.37	0.54	0.52	0.52	
	MLP	0.23	0.40	0.44	0.44	
NASNet Mobile	LSVM	0.68	0.70	0.68	0.67	
	CNN (1D)	0.74	0.77	0.73	0.71	
	GBDT	0.55	0.52	0.51	0.54	
	SGD	0.58	0.68	0.61	0.64	
	MLP	0.39	0.63	0.08	0.08	
MobileNet V2	LSVM	0.69	0.70	0.70	0.68	
	CNN (1D)	0.82	0.79	0.79	0.73	
	GBDT	0.07	0.42	0.42	0.56	
	SGD	0.68	0.65	0.68	0.65	
	MLP	0.34	0.60	0.08	0.08	
AVERAGE		**0.57**	**0.62**	**0.55**	**0.57**	

Overall, PCA did not enhance classification accuracy considerably. SGD and MLP are the only two classifiers that performed better with PCA. For instance, EFFNET-SGD accuracy increased from 0.68 to 0.94, whereas NASNETMobile's classification accuracy increased from 0.39 to 0.63.

The accuracy performance of LDA and BORUTA treatment as compared to without feature selection treatment (Baseline) can be referred in Table XVIII. As depicted in Table XVIII, LDA performed excellently on many features and classifiers, except EFFNET-GBDT, NASNETMobile-GBDT and MOBILENetV2-GBDT. In contrast, BORUTA did not increase the accuracy of nearly all features, and there was even a slight drop in accuracy.

TABLE XVIII. EFFECTS OF LDA AND BORUTA ON ACCURACY FOR SCENE-15 DATASET

Feature Extraction	Classification	Baseline	LDA	BORUTA	Spark line
EFFNet	LSVM	0.94	0.99	0.93	
	CNN (1D)	0.94	0.99	0.93	
	GBDT	0.68	0.05	0.70	
	SGD	0.68	0.99	0.90	
	MLP	0.43	0.69	0.64	
RESNet152	LSVM	0.62	0.93	0.58	
	CNN (1D)	0.62	0.93	0.60	
	GBDT	0.41	0.69	0.38	
	SGD	0.37	0.94	0.36	
	MLP	0.23	0.58	0.08	
NASNet Mobile	LSVM	0.68	0.91	0.40	
	CNN (1D)	0.74	0.90	0.72	
	GBDT	0.55	0.04	0.55	
	SGD	0.58	0.89	0.42	
	MLP	0.39	0.77	0.46	
MobileNetV2	LSVM	0.69	0.75	0.62	
	CNN (1D)	0.82	0.91	0.79	
	GBDT	0.07	0.01	0.19	
	SGD	0.68	0.91	0.61	
	MLP	0.34	0.75	0.23	
AVERAGE		**0.57**	**0.73**	**0.55**	

The analysis of RFE accuracy performance is shown in Table XIX. The pattern of data presented in Table XIX obviously indicates that RFE has brought less impact on improving almost all feature representation. However, the positive effects of RFE can be seen on EFFNET-SGD, EFFNET-MLP and MobileNetV2-MLP.

TABLE XIX. EFFECTS OF RFE ON ACCURACY FOR SCENE-15 DATASET

Feature Extraction	Classification	Baseline	RFE 1	RFE 2	RFE 3	Spark line
EFFNet	LSVM	0.94	0.94	0.93	0.89	
	CNN (1D)	0.94	0.92	0.93	0.89	
	GBDT	0.68	0.61	0.70	0.66	
	SGD	0.68	0.91	0.92	0.84	
	MLP	0.43	0.54	0.36	0.38	
RESNet152	LSVM	0.62	0.59	0.46	0.04	
	CNN (1D)	0.62	0.57	0.53	0.11	
	GBDT	0.41	0.40	0.37	0.12	
	SGD	0.37	0.34	0.45	0.08	
	MLP	0.23	0.24	0.08	0.08	
NASNet Mobile	LSVM	0.68	0.19	0.04	0.04	
	CNN (1D)	0.74	0.79	0.63	0.20	
	GBDT	0.55	0.66	0.56	0.07	
	SGD	0.58	0.44	0.08	0.06	
	MLP	0.39	0.27	0.08	0.08	
MobileNet V2	LSVM	0.69	0.70	0.68	0.06	
	CNN (1D)	0.82	0.80	0.80	0.60	
	GBDT	0.07	0.07	0.07	0.19	
	SGD	0.68	0.67	0.69	0.23	
	MLP	0.34	0.59	0.08	0.08	
AVERAGE		**0.57**	**0.56**	**0.47**	**0.28**	

Fig. 17 to 20 show a detailed analysis of feature selection performance for each feature and machine learning classifier. Based on the graph shown in Fig. 17, the transformation of EFFNET feature by using LDA had improved the classification accuracy of LSVM, CNN, SGD and MLP. In addition to that, the PCA, BORUTA and RFE brought significant effects on the accuracies for MLP and SGD.

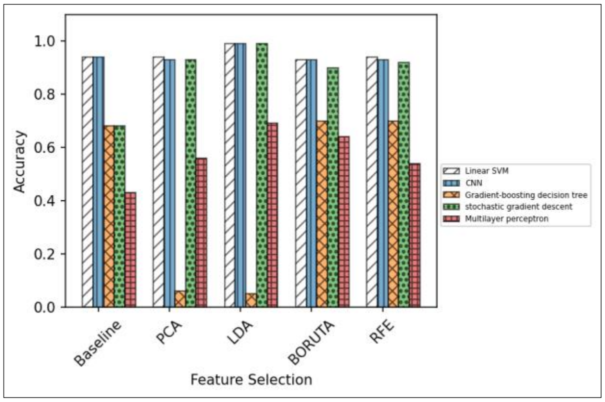


Fig. 17. Effects of feature selection on machine learning classifiers for EFFNET (scene-15 dataset)

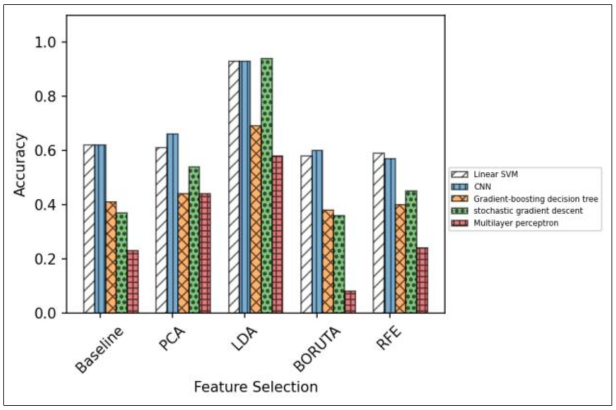


Fig. 18. Effects of feature selection on machine learning classifiers for RESNET152 (scene-15 dataset)

As for RESNET152, as shown in Fig. 18, there was a tremendous increase on the accuracy when LDA was being used to transform the features for CNN, LSVM and SGD. The rest of the feature selection techniques by using PCA, BORUTA and RFE seemed to have less positive impacts on the accuracies. Similarly in Fig. 19 and Fig. 20, LDA still outperformed the accuracy of PCA, BORUTA and RFE on all classifiers except GBDT. For NASNETMobile, PCA demonstrated a bit of an improvement on the accuracies for CNN, SGD and MLP. There were no positive effects on the LSVM, CNN, SGD, and MLP accuracies for BORUTA and RFE.

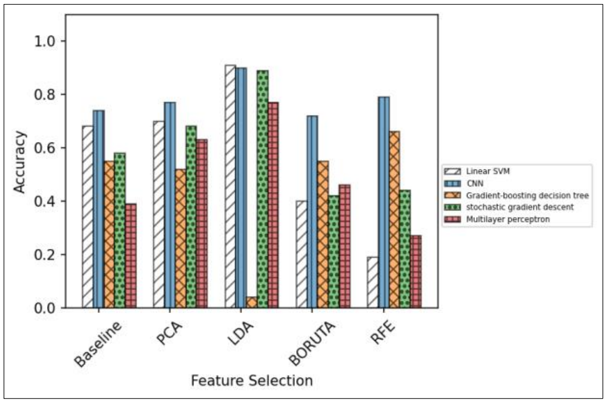


Fig. 19. Effect of feature selection on machine learning classifiers for NASNETMOBILE (scene-15 dataset)

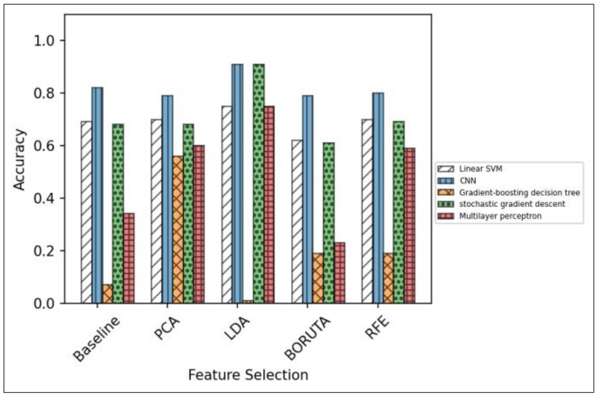


Fig. 20. Effect of feature selection on machine learning classifiers for MOBILENETV2 (scene-15 dataset)

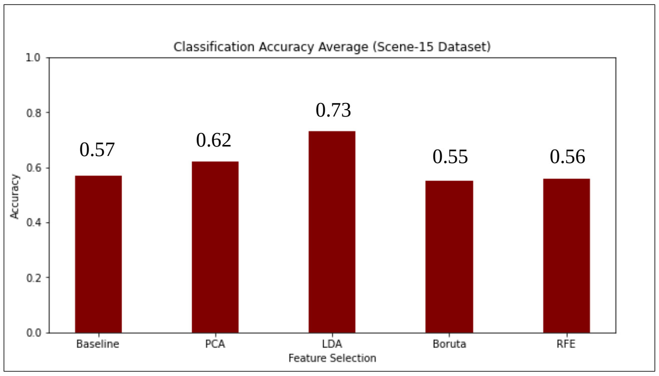


Fig. 21. The average performance comparisons of feature selection for scene-15 dataset

Fig. 21 shows the summary of feature selection performance on the Scene-15 dataset. LDA was the best feature selection technique for the Scene-15 dataset since it not only worked with a wide range of features and classifiers, but it also improved classification accuracy significantly. BORUTA and RFE, on the other hand, have no substantial impact on classification performance. Due to the constant performance across numerous feature selections, it can also be inferred that EFFNET is the best features and, LSVM is the best classifier.

V. Conclusion and Future Works

This paper evaluated several transfer learning approaches and feature selections for effective and super lightweight landmark recognition model. A landmark recognition model was trained through the features extraction by using the pre-trained CNN architectures and machine learning classifiers. The new UMS landmark datasets were created, and the landmark recognition model was also evaluated with the Scene-15 dataset. The findings showed that the EFFNET CNN architecture with CNN classifier is the best feature extraction and classifier in this study. EFFNET-CNN achieved 100% and 94.26% accuracies on UMS landmark and Scene-15 dataset, respectively. Moreover, the features created by EFFNET were more compact compared to the other features. Furthermore, based on the evaluation of several feature selection algorithms, LDA was determined to be the best feature selection technique for vastly reducing feature dimensionality by 99.69% for UMS landmark dataset and 98.90% for Scene-15 dataset while maintaining good accuracies. However, although a super lightweight landmark recognition model was produced, it must undergo extra pre-processing step to reduce the dimensionality of features which will impose excessive computational costs of processing. Therefore, future works that can be suggested are to evaluate the effect of the proposed dimensionality reduction technique on the computational cost of the algorithms as well as to test it on various benchmark datasets.

References

[1] L. Xie, F. Lee, L. Liu, K. Kotani, and Q. Chen, “Scene recognition: A comprehensive survey,” Pattern Recognit., vol. 102, 2020, doi: 10.1016/j.patcog.2020.107205.

[2] A. Boiarov and E. Tyantov, “Large scale landmark recognition via deep metric learning,” Int. Conf. Inf. Knowl. Manag. Proc., pp. 169–178, 2019, doi: 10.1145/3357384.3357956.

[3] M. Jiafa, W. Weifeng, H. Yahong, and S. Weiguo, “A scene recognition algorithm based on deep residual network,” Syst. Sci. Control Eng., vol. 7, no. 1, pp. 243–251, 2019, doi: 10.1080/21642583.2019.1647576.

[4] N. M. Firdaus, D. Chahyati, and M. I. Fanany, “Tourist attractions classification using ResNet,” 2018 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2018, pp. 429–433, 2019, doi: 10.1109/ICACSIS.2018.8618235.

[5] M. Jiafa, W. Weifeng, H. Yahong, and S. Weiguo, “A scene recognition algorithm based on deep residual network,” Syst. Sci. Control Eng., vol. 7, no. 1, pp. 243–251, 2019, doi: 10.1080/21642583.2019.1647576.

[6] M. N. Razali, A. S. Shafie, and R. Hanapi, “Performance Evaluation of Masked Face Recognition Using Deep Learning for Covid-19 Standard of Procedure (SOP) Compliance Monitoring,” 2021 6th IEEE Int. Conf. Recent Adv. Innov. Eng. ICRAIE 2021, vol. 2021, 2021, doi: 10.1109/ICRAIE52900.2021.9703986.

[7] M. N. Razali, A. S. Shafie, and R. Hanapi, “Performance Evaluation of Masked Face Recognition Using Deep Learning for Covid-19 Standard of Procedure (SOP) Compliance Monitoring,” vol. 2021, pp. 1–7, 2022, doi: 10.1109/icraie52900.2021.9703986.

[8] [8] S. Khan et al., “DeepSmoke: Deep learning model for smoke detection and segmentation in outdoor environments,” Expert Syst. Appl., vol. 182, no. December 2020, p. 115125, 2021, doi: 10.1016/j.eswa.2021.115125.

[9] E. Uçar, Ü. Atila, M. Uçar, and K. Akyol, “Automated detection of Covid-19 disease using deep fused features from chest radiography images,” Biomed. Signal Process. Control, vol. 69, no. December 2020, p. 102862, 2021, doi: 10.1016/j.bspc.2021.102862.

[10] N. Jahan, M. S. Anower, and R. Hassan, “Automated Diagnosis of Pneumonia from Classification of Chest X-Ray im ages using EfficientNet,” 2021 Int. Conf. Inf. Commun. Technol. Sustain. Dev. ICICT4SD 2021 - Proc., pp. 235–239, 2021, doi: 10.1109/ICICT4SD50815.2021.9397055.

[11] M. M. A. Monshi, J. Poon, V. Chung, and F. M. Monshi, “CovidXrayNet: Optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR,” Comput. Biol. Med., vol. 133, no. March, p. 104375, 2021, doi: 10.1016/j.compbiomed.2021.104375.

[12] A. A. Pokroy and A. D. Egorov, “EfficientNets for DeepFake Detection: Comparison of Pretrained Models,” Proc. 2021 IEEE Conf. Russ. Young Res. Electr. Electron. Eng. ElConRus 2021, pp. 598–600, 2021, doi: 10.1109/ElConRus51938.2021.9396092.

[13] A. S. Timmaraju and A. Chatterjee, “Monulens : Real-time mobile-based Landmark Recognition.”

[14] Meiliana, D. Irmanti, M. R. Hidayat, N. V. Amalina, and D. Suryani, “Mobile Smart Travelling Application for Indonesia Tourism,” Procedia Comput. Sci., vol. 116, pp. 556–563, 2017, doi: 10.1016/j.procs.2017.10.059.

[15] A. Crudge, W. Thomas, and K. Zhu, “Landmark Recognition Using Machine Learning,” pp. 1–5, 2014.

[16] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, “Learning deep features for scene recognition using places database,” Adv. Neural Inf. Process. Syst., vol. 1, no. January, pp. 487–495, 2014.

[17] N. Shigei, K. Mandai, S. Sugimoto, R. Takaesu, and Y. Ishizuka, “Land-use classification using convolutional neural network with bagging and reduced categories,” Lect. Notes Eng. Comput. Sci., vol. 2239, pp. 7–11, 2019.

[18] E. G. Moung, C. J. Hou, M. M. Sufian, M. H. A. Hijazi, J. A. Dargham, and S. Omatu, “Fusion of moment invariant method and deep learning algorithm for COVID-19 classification,” Big Data Cogn. Comput., vol. 5, no. 4, 2021, doi: 10.3390/bdcc5040074.

[19] V. Parikh, M. Keskar, D. Dharia, and P. Gotmare, “A Tourist Place Recommendation and Recognition System,” Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2018, no. Icicct, pp. 218–222, 2018, doi: 10.1109/ICICCT.2018.8473077.

[20] UMS, “Landmarks in UMS,” 2021. https://www.ums.edu.my/v5/en/landmark-of-ums.

[21] E. O. Nixon and M. N. Razali, “Ums Landmark Recognition Dataset,” 2022. https://doi.org/10.34740/KAGGLE/DS/1877538.

[22] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2, pp. 2169–2178, 2006, doi: 10.1109/CVPR.2006.68.

[23] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” 36th Int. Conf. Mach. Learn. ICML 2019, vol. 2019-June, pp. 10691–10700, 2019.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[25] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning Transferable Architectures for Scalable Image Recognition,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 8697–8710, 2018, doi: 10.1109/CVPR.2018.00907.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[27] M. M. A. Monshi, J. Poon, V. Chung, and F. M. Monshi, “CovidXrayNet: Optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR,” Comput. Biol. Med., vol. 133, no. December 2020, p. 104375, 2021, doi: 10.1016/j.compbiomed.2021.104375.

[28] M. N. Razali et al., “Indigenous food recognition model based on various convolutional neural network architectures for gastronomic tourism business analytics,” Inf., vol. 12, no. 8, 2021, doi: 10.3390/info12080322.

[29] V. K. Shrivastava, M. K. Pradhan, and M. P. Thakur, “Neural Networks for Rice Plant Disease Classification,” pp. 1023–1030, 2021.

[30] C. Cordoș, L. Mihailă, P. Faragó, and S. Hintea, “ECG signal classification using Convolutional Neural Networks for Biometric Identification,” pp. 167–170, 2021.

[31] R. V. M. Da Nóbrega, S. A. Peixoto, S. P. P. Da Silva, and P. P. R. Filho, “Lung Nodule Classification via Deep Transfer Learning in CT Lung Images,” Proc. - IEEE Symp. Comput. Med. Syst., vol. 2018-June, pp. 244–249, 2018, doi: 10.1109/CBMS.2018.00050.

[32] S. A. A. Ahmed, B. Yanikoglu, O. Goksu, and E. Aptoula, “Skin Lesion Classification with Deep CNN Ensembles,” 2020 28th Signal Process. Commun. Appl. Conf. SIU 2020 - Proc., pp. 1–4, 2020, doi: 10.1109/SIU49456.2020.9302125.

[33] N. Merrin Prasanna, D. Subash Chandra Mouli, G. Sireesha, K. Priyanka, D. Radha, and B. Manmadha, “Classification of food categories and ingredients approximation using an fd-mobilenet and TF-Yolo,” Int. J. Adv. Sci. Technol., vol. 29, no. 6, pp. 3104–3114, 2020.

[34] M. N. S. Zainudin, N. Sulaiman, N. Mustapha, T. Perumal, and R. Mohamed, “Two-stage feature selection using ranking self-adaptive differential evolution algorithm for recognition of acceleration activity,” Turkish J. Electr. Eng. Comput. Sci., vol. 26, no. 3, pp. 1378–1389, 2018, doi: 10.3906/elk-1709-138.

[35] L. P. Hung, R. Alfred, and M. H. A. Hijazi, “Comparison of feature selection methods for sentiment analysis,” Commun. Comput. Inf. Sci., vol. 872, no. February 2020, pp. 261–272, 2018, doi: 10.1007/978-3-319-96292-4_21.

[36] M. N. Razali, N. Manshor, A. A. Halin, N. Mustapha, and R. Yaakob, “Extremal Region Selection for MSER Detection in Food Recognition,” ASM Sci. J., vol. 15, no. May, pp. 1–11, 2021, doi: 10.32802/ASMSCJ.2020.485.

[37] D. J. Bartholomew, “Principal components analysis,” Int. Encycl. Educ., pp. 374–377, 2010, doi: 10.1016/B978-0-08-044894-7.01358-0.

[38] F. Song, D. Mei, and H. Li, “Feature selection based on linear discriminant analysis,” Proc. - 2010 Int. Conf. Intell. Syst. Des. Eng. Appl. ISDEA 2010, vol. 1, pp. 746–749, 2010, doi: 10.1109/ISDEA.2010.311.

[39] M. B. Kursa, A. Jankowski, and W. R. Rudnicki, “Boruta - A system for feature selection,” Fundam. Informaticae, vol. 101, no. 4, pp. 271–285, 2010, doi: 10.3233/FI-2010-288.

[40] X. Zeng, Y. W. Chen, C. Tao, and D. Van Alphen, “Feature selection using recursive feature elimination for handwritten digit recognition,” IIH-MSP 2009 - 2009 5th Int. Conf. Intell. Inf. Hiding Multimed. Signal Process., pp. 1205–1208, 2009, doi: 10.1109/IIH-MSP.2009.145.