APPLIED MACHINE LEARNING SYSTEM ELEC0134 22/23 REPORT

SN: 22039170

ABSTRACT

There are two categories of tasks raised in this mini project. They are binary classification and multiclass classification for image datasets. Machine learning techniques including convolutional neural network (CNN), support vector machine (SVM), random forest (RF) and residual network (ResNet) are applied to address the image classification tasks. In this report, those machine learning models mentioned above are illustrated and detailed implementation procedure for each model is explained. Experimental results are described, and it is noticed that neural network (NN) based models including CNN and ResNet performs better than the other two models with the accuracy of 0.7550 for CNN and 0.8064 for ResNet respectively.

Index Terms— Machine learning, image classification, neural networks

1. INTRODUCTION

This report describes the procedure of solving machine learning tasks including A1, A2, B1 and B2 among which A1 and A2 are Binary classification tasks and B1 and B2 are Multiclass classification tasks. Specifically, task A1 requires a machine learning model for gender detection given an image to determine if the person in it is male or female. Similarly, task A2 is aimed at training a machine learning model to emotion detection given an image to determine if the person in it is smiling or not. In the contrast, task B1 and B2 require machine learning model which could perform a classification task for 5 classes which is more than two (binary) classes. Convolutional neural network (CNN), support vector machine (SVM), random forest (RF) and residual network (ResNet) are applied to task A1, A2, B1 and B2 respectively. They are all belong to supervised machine learning methods but CNN and RF are deep learning based methods while SVM and RF are not.

This report will first provide a literature review of machine learning methods including CNN, SVM, RF, ResNet and other classic or state-of-the-art algorithms. The review will explain which machine learning problem they could tackle with. Then it will give a relatively brief description of specific models using in the project and the reason of choosing them. In addition, this report will elaborate detailed implementation of machine learning models, including description of given datasets and separation criterion. Also, key algorithms, modules and training pipelines are illustrated in this section, as well as thorough discussion on the learning curve and training convergence. It is followed by the experimental results and analysis including discussion of model prediction accuracy. Finally, this report ends with a conclusion of what is found in this project and what could be improved or adjusted for this project. The corresponding code is public on github.[1]

2. LITERATURE SURVEY

SVM is one of the most classic classification algorithms with high generalization ability. It could classify object to one class without prior knowledge after SVM classifier is trained. Its core intuition is to separate two groups of data with a hyperplane and keep the distance between data points in each group and the hyperplane as far as possible. By mapping the linearly non-separable data into higher dimensions, SVM could find a separator to classify two groups of data. It is proved possible to apply SVM to image classification tasks by Chapelle, et al. [1] whose work achieves 11% error rate. Random forests algorithm is another classic classification algorithm. As its name suggests, it is a combination of decision trees whose classification results are collected and voted to form a majority which is the predicted class. The generalization error of RF becomes small with the number of trees increase. It is also affected by the strength of individual decision tree and the correlation between trees [2].

On the other hand, CNN comes with the start of deep learning era. It is basically a neural network architecture with the assistance of convolution operation and pooling operation. Non-linear activation layers are also important component because it could be used to adjust or cut-off the output from former layers [3]. Fully connected layers in CNN perform as in plain neural networks a similar function which is directly connecting every single node in former layers and conducting calculations for predicting. VGG net developed by Simonyan and Zisserman [4] achieved state-of-the-art performance at ICLR 2015. In particular, they conduct a thorough evaluation of how using small convolution filters like 3 by 3 filters thereby making the network deeper could improve the performance of the architecture. VGG net

[1] The code is provided link-to-download-your-project.com and GitHub project: https://github.com/N1ghtstalker2022/AMLS_assignment22_23.git

outperform almost all other available models in localization and classification tasks specified by ImageNet Challenge 2014. ResNet proposed by He, et al. [5] beat VGG net and could have several times more layers without increasing model complexity. It is proved to be suitable for detection and localization tasks.

3. DESCRIPTION OF MODELS

3.1. Task A: Binary tasks

3.1.1 Gender detection

For this gender detection task, CNN model is used to perform binary classification. CNN follows the basic structure of neural networks which is composed of input layer, hidden layer, and output layer. The hidden layer could contain convolutional layers, pooling layers, and fully connection layers.

Basically, the convolutional layer is a processing kernel which perform convolution operation to the input. Convolutional layers receive tensors with a shape like with a shape: (number of inputs) × (input height) × (input width) × (input channels). Then the convolutional layer convolves the input and passes it to the next layer. The convolution operation could generate new feature map given specific input and is usually followed by a pooling operation which is performed in pooling layers. In pooling layers, the pooling operation is commonly done by either max pooling or average pooling which combines local clusters like 2 by 2 matrix to a single value (maximum number or average).

Both convolutional layers and pooling layers are critical parts of CNN because they both vastly reduce the parameters of the whole neural network. This is a rather excellent characteristics for tasks with large amount of input data like image classification problems. For example, a 128 by 128 image could have 16384 input data points which requires 16384 weights in the second layer for a fully connected neural network, but with convolution the number of weights (parameters) could be reduced drastically because in CNN many neurons in a certain convolutional layer could share the same convolution kernel and with pooling operation number of parameters are further reduced to some extent. Moreover, with reduced number of parameters, CNN could be trained with less time and require less memory to store the parameters, which means less computational costs.

3.1.1 Emotion detection

For this emotion detection task, SVM is used to classify whether the human in a given image is smiling or not. SVM is one of the most robust machine learning methods based on statistical learning frameworks. The core idea of SVM training is trying to map training examples to different categories in order to maximize the gap between two categories. SVM could not only perform linear classification, but also non-linear classification thanks to kernel trick which means implicitly mapping data in low dimensional space to high dimensional feature space. Previous experiments have shown SVM could be used to image classification problems and could have decent predicting accuracy.

Here Non-linear SVM is applied to perform image classification but basic theory of linear SVM is first introduced in this report due to its easiness of demonstration.

Let x_i be the i_{th} input image matrix where $i = 1, 2, \dots, n$ and n is the total number of input images. y_i is the corresponding label for the i_{th} input image where y_i is either 1 or -1 indicating the label. The given training dataset could be written as $(x_1, y_1), \dots, (x_n, y_n)$. SVM aims at finding the maximum margin hyperplane which devides given images to two distinct categories with its label y_i being either 1 or -1. The case of maximum margin hyperplane would be satisfied when the distance between the hyperplane and the most adjacent data x_i from each category is maximized. The hyperplane could be denoted using the following equation: $w^T x - b = 0$, where w is the normal vector of the hyperplane. Since it is assumed that training data is linearly separable. Two parallel hyperplanes could be selected to separate the input data to two categories. The margin is the region bounded by these two hyperplanes which could be written as: $w^T x - b = 1$ and $w^T x - b = -1$. By mathematical formula of lines, the distance between these two hyperplanes is $\frac{2}{||w||}$. In order to let each data point fall into pre-defined clusters, the following constraint is needed: $w^T x - b \geq 1, if\ y_i = 1$, and $w^T x - b \leq -1, if\ y_i = -1$. These two constrains state that each data point should be classified to its corresponding correct category and could be rewritten as $y_i(w^T x - b) \geq 1$. Here we obtain the optimization problem which is $minimize\ \left||w|\right|_2^2, subject\ to\ y_i(w^T x - b) \geq 1.$ Since image data is far more completed from data perspective, it is impossible to be linearly separated. Hence, hinge loss function is applied to calculate the loss. We obtain the following $max\big(0,1 - y_i(w^T x - b)\big)$, which gives zero if an image is correctly classified, i.e. x_i lies on the correct side of the margin and gives proportionally increasing value with respect to the distance from the margin for data on the wrong side of the margin. The optimization problem then becomes to minimize $\lambda\left||w|\right|_2^2 + \left[\frac{1}{n}\sum_{i=1}^{n} max\big(0,1 - y_i(w^T x - b)\big)\right]$, where the parameter λ determines the trade-off between increasing the margin size and ensuring that the data point lie on the correct side of the margin.

Furthermore, kernel trick could be applied to solve the optimization problem, which means the algorithm could update its parameters. In another word, SVM becomes trainable and could be applied to classification task.

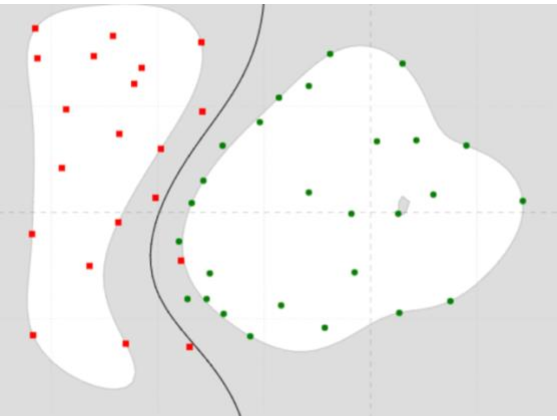

Fig. 1 SVM model

3.2. Task B: Multiclass tasks

3.2.1 Face shape recognition

For this multiclass classification task, Random Forest algorithm (RF) is used. It is basically an ensemble learning methods constructing a group of decision trees at training time. Ensemble learning methods combine several machine learning techniques into one predictive model in order to decrease variance and/or bias. Random forests are examples of ensemble methods.

RF will choose the class voted by most decision trees for classification tasks. machine learning models such as decision trees are simple but often overfit leading to weak noisy predictors. Overfitting is associated with low-bias models like single decision tree which could lead to high variance (in the sense that training is more susceptible to the influence of examples in the training set). Forests are like the pulling together of decision tree algorithm efforts. Taking the teamwork of many trees thus improving the performance of a single random tree.

Bagging can help by decreasing variance of the base model without influencing bias but can also result in loss of simple structures in the original model. Concretely, the method of bagging implemented in RF could be illustrated as follows. First during training process, we set the base model as decision tree and sample with replacement n groups of training data fed into the same number of trees. Then after training, classification of a new given data could be made by majority voting which means find the majority of classification result given by n decision trees.

The rough RF model could be described in Fig. 2. This RF model comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

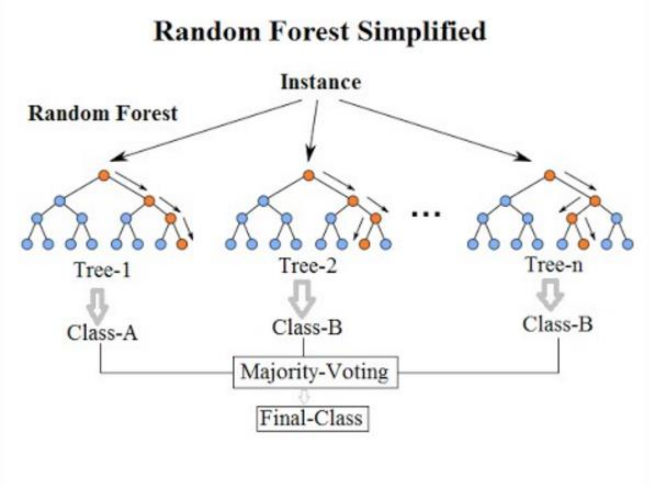

Fig. 2 Random Forest model

3.2.2 Eye color recognition

For this eye color recognition task, ResNet-34 is used to build the base model. ResNet is a special type of Convolutional Neural Network (CNN) that is used for tasks like Image Recognition. A ResNet can be called an upgraded version of the VGG architecture, with the difference between them being the skip connections used in ResNets. To obtain better predicting accuracy, we need deeper neural network, but the performance of Pure CNN becomes worse with the increasement of depth of neural network. This is counterintuitive because the training error of deeper network should be lower than that of shallow one rather than higher. It could be because the vanishing gradient problem whose solution is to use skip connection. This vanishing gradient problem suggest that the optimal mapping might be closer to identity mapping x rather than the feature mapping for x [5]. Thus by adding identity block x to the output, it helps precondition the network in case the optimal function is closer to the identity mapping rather than a zero mapping performed by intermediate feature mapping layers.

As it is demonstrated in Fig. 3, the skip connection means adding the residue to the results calculated by convolution and batch normalization and then generate the output fed to the following layers. The left structure in Fig. 3 is called Identity block in which the residue is input x itself, while the right one is called bottleneck block in which the residue is input x after a 1 by 1 convolution block.

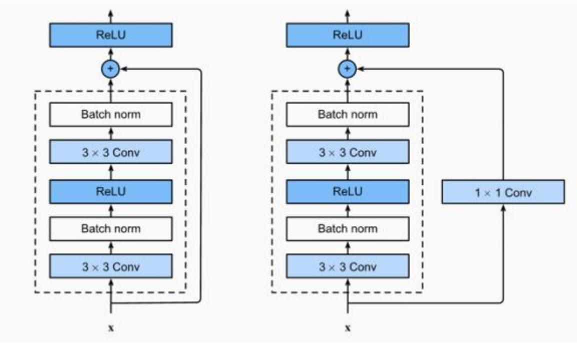

Fig. 3 ResNet skip connection

The whole ResNet architecture containing core building block of skip connection could be succinctly represented in Fig. 4. It could be illustrated as a network architecture composed of embedding, mapping and prediction layer, where convolution layers are used in mapping to extract the features presented in images and fully connected layers are used in prediction.

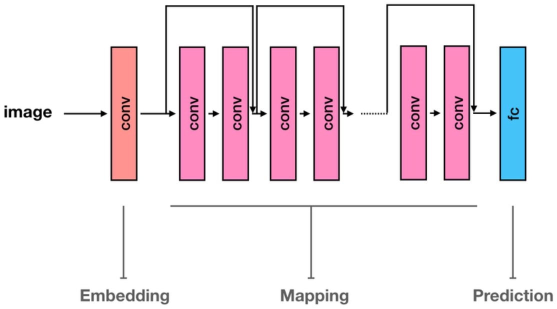

Fig. 4 ResNet architecture

4. IMPLEMENTATION

4.1. Task A: Binary tasks

The dataset for gender detection contains 6000 images with corresponding labels. Having a glimpse of images, it is noticed that gender label 1 denotes men and label -1 denotes women. Smiling label 1 denotes person in the image is smiling and label -1 denotes person in the image is not smiling.

It is noteworthy that the size of each image is 178 by 218 which is a bit large. To reduce the computation cost, we down sampling it into 50 by 50 images. Then, we rescale the pixel value for each given image from $(0, 255)$ to a range of $(0, 1)$. This is an important step which ensures that each input parameter (pixel, in this case) has a similar data distribution. This makes convergence faster while training the network. The data is split into train, validation, and test sets with a ratio of 4: 1: 1.

K-fold cross validation method is used to choose the better hyperparameter in this case learning rate of model optimizer. There is a group of learning rate to be selected. K-fold cross validation first split the training set into K groups (in this task K=5). Then in a Round-Robin fashion, one of the split datasets is used as validation set while others are training set and then model is trained on training sets and model performance is tested on validation sets. This process repeats K times and generate a model performance measured by the average of K tested performance. This model performance is compared with all the other performance with different learning rate. Finally, the best learning rate is chosen to be used for formal model training process.

4.1.1 Gender detection

Tensorflow and keras are used in the code to build CNN model. More specifically, the exact CNN model used for this task is shown in Fig. 5. This model contains three convolution layers (3 by 3 kernel) with three downstream pooling layers (2 by 2 maxpooling) and two fully connected layers. The output is an integer representing the class of the image.

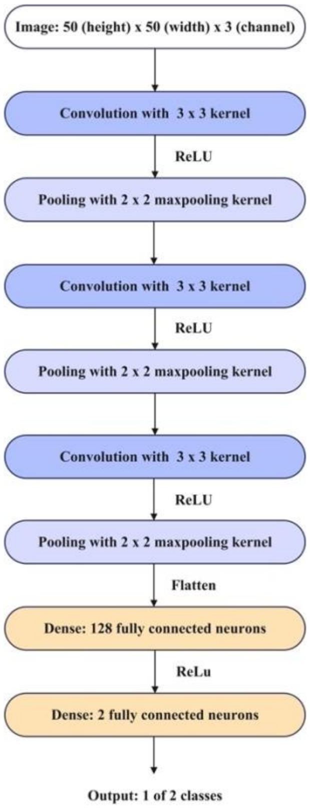
Image: 50 (height) x 50 (width) x 3 (channel)
Convolution with 3 x 3 kernel
ReLU
Pooling with 2 x 2 maxpooling kernel
Convolution with 3 x 3 kernel
ReLU
Pooling with 2 x 2 maxpooling kernel
Convolution with 3 x 3 kernel
ReLU
Pooling with 2 x 2 maxpooling kernel
Flatten
Dense: 128 fully connected neurons
ReLu
Dense: 2 fully connected neurons
Output: 1 of 2 classes

Fig. 5 CNN model

Learning curve including training and validation accuracy and loss are plotted using matplotlib module in python. As it could be seen in the learning curve graph Fig. 6 and Fig. 7, both training and validation accuracy of the CNN model with

K-fold cross validation is much better than the CNN model without cross validation. Training and validation loss of the CNN model with K-fold cross validation is much lower than the one without cross validation. Also, for CNN with cross validation, both the training and validation accuracy are increasing with the increase of epochs and in the end of training the training accuracy is close to validation accuracy, which means there could be very little overfitting problems.

The learning curve also shows the importance of choosing learning rate. With good learning rate the CNN model could converge more quickly and obtain a competence of validating with high accuracy, whereas with a not decent learning rate the model will perform not as it is wished and be difficult to converge.

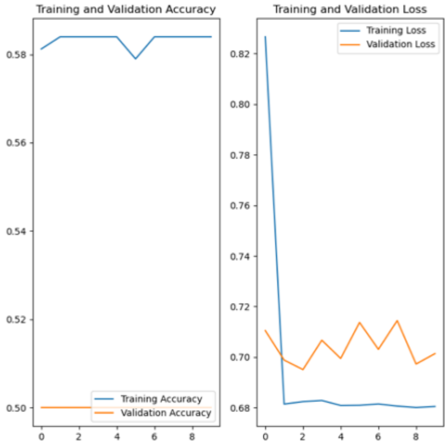

Fig. 6 Learning curve for CNN

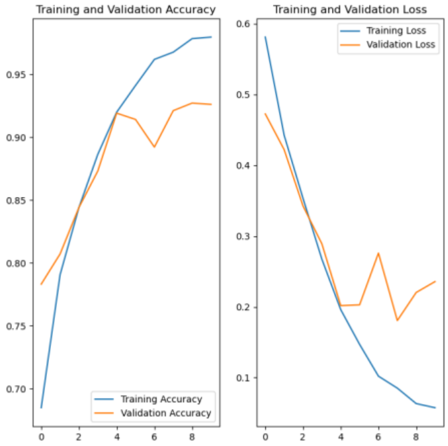

Fig. 7 Learning curve for CNN with cross validation

4.1.2 Emotion detection

In this task, scikit-learn is used to build SVM classifier model. The SVC class provides C-support vector classification function in which there are several tunable parameters including Regularization parameter C, kernel type, and kernel coefficient gamma to be specified.

SVM is defined as a mathematical model with a number of parameters that need to be learned from the data. However, there are some parameters, known as Hyperparameters and those cannot be directly learned. They are commonly chosen by humans based on some intuition or hit and trial before the actual training begins. Models can have many hyper-parameters and finding the best combination of parameters can be treated as a search problem.

SVM has some hyper-parameters and finding optimal hyper-parameter is a very hard task to solve. But it can be found by just trying all combinations and see what parameters work best. The main idea behind it is to create a grid of hyper-parameters and try all of their combinations. Scikit-learn has a functionality built-in with GridSearchCV which takes a dictionary that describes the parameters that could be tried on a model to train it. The grid of parameters is defined as a dictionary, where the keys are the parameters and the values are the settings to be tested. Through the Grid Search cross validation, the model which performs best with its corresponding parameters is chosen as the best estimator.

Standard SVM are convex optimization problems, therefore the algorithm will always converge if it reaches global maxima.

Learning curve is displayed in Fig. 8.

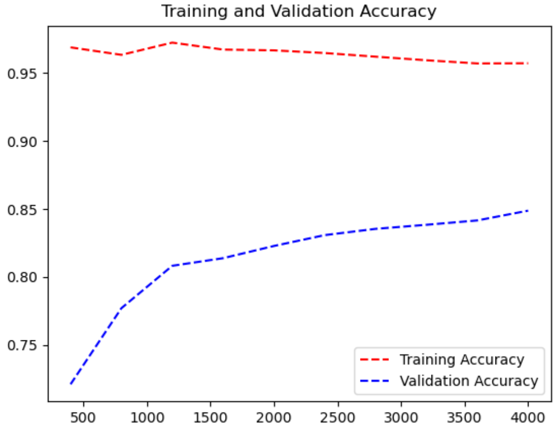

Fig. 8 Learning curve for SVM

4.2. Task B: Multiclass tasks

The dataset for this task is also composed of 6000 images with corresponding labels (eye color or face shape). The digital number from 0 to 4 represents different eye color feature and face shape feature.

It is noteworthy that the size of each image is 500 by 500 which is a bit large. To reduce the computation cost, we down sampling it into 50 by 50 images. Then, we rescale the pixel value for each given image from $(0, 255)$ to a range of $(0, 1)$.

This is an important step which ensures that each input parameter (pixel, in this case) has a similar data distribution. This makes convergence faster while training the network. The data is split into train, validation, and test sets with a ratio of 4: 1: 1.

4.2.1 Face shape recognition

Random Forest (RF) model is used to carry out face shape recognition task. In this task, the python module called tensorflow_decision_forests is used as the base RF model. There are 300 decision trees in the RF and the training data with a batch size of 256 is fed into these 300 decision trees.

The training logs show the quality of the model (e.g. accuracy evaluated on the out-of-bag dataset) according to the number of trees in the model. These logs are helpful to study the balance between model size and model quality.

The following plots in Fig. 9 show the accuracy and losses of out-of-bag datasets which could be treated as validation dataset during training.

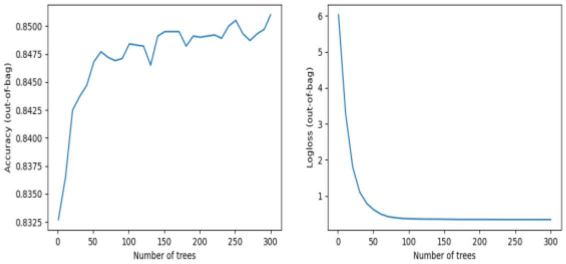

Fig. 9 Learning curve of random forest

4.2.2 Eye color recognition

Resnet-34 model is used to carry out eye color recognition task. Model construction is implemented by tensorflow. As it is shown in Fig. 10, the main building block for resnet-34 is four types of convolution layers which have the same filter size 3 by 3 but with different filter number. As a matter of fact, one conv block is composed of convolution, batch norm and activation operation.

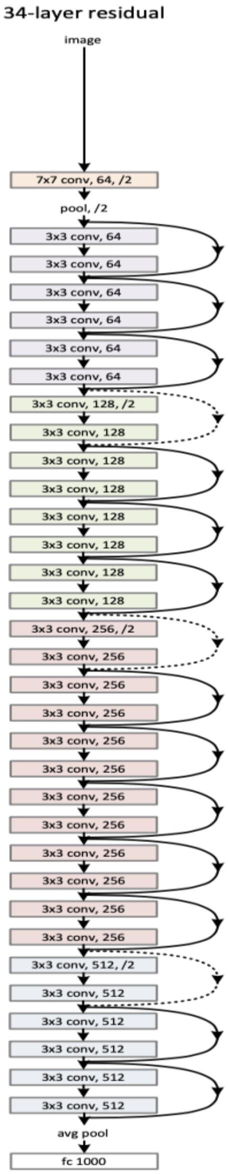

Fig. 10 Resnet-34 implementation

Scikit-learn module in python is used for k-fold cross validation. The Adam optimizer is chosen, and the suitable learning rate is obtained by cross validation procedure.

The learning curve in Fig. 11also shows how the training and validation accuracy increases and loss decreases after each epoch. The result could be considered as decent because after 10 epochs, because both the training and validation accuracy are high despite validation accuracy falls slightly behind training accuracy, indicating low bias and low variance of the model performance.

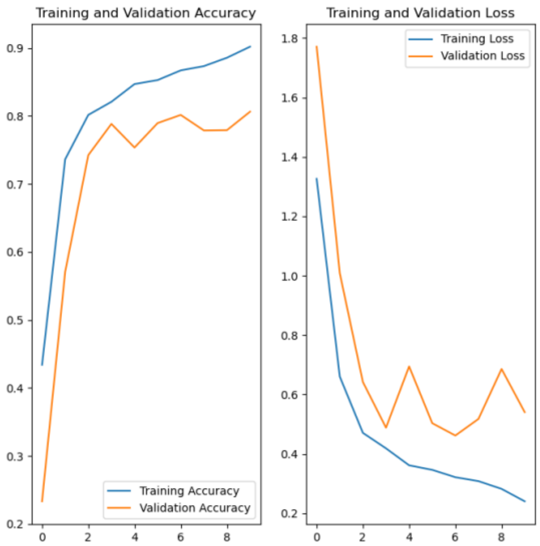

Fig. 11 Learning curve for ResNet with learning rate 1e-4

5. EXPERIMENTAL RESULTS AND ANALYSIS

After model implementation and hyperparameter tunning, the model with expected performance is obtained. The model performance is tested by using testing dataset as the model input.

Table 1 shows the corresponding testing accuracy for CNN, SVM, RF, ResNet which is used in task A1, A2, B1 and B2 respectively.

It is noteworthy that neural network (NN) based model like CNN and ResNet performs better than non-NN model like SVM and RF. The possible reason for this might be the very high dimension input features. Although the image is resized to 50 by 50 before input into the model, the number of features is still as large as 2500. Model like SVM and RF might not be capable of generalizing as well as NN based models.

Therefore, for tasks like image classification, choosing NN based model could be an expedient of obtaining better results.

Table 1. Testing accuracy of models for each task

Task List	Model	Accuracy
A1	CNN	0.7550
A2	SVM	0.6800
B1	RF	0.7920
B2	ResNet	0.8064

Also, it should be mentioned that the training time for ResNet is much longer than CNN model. The explosive increase of trainable parameters could be a crucial reason. The ResNet-34 model has as many as 34 convolution layers compared with shallow CNN model with 5 convolution layers. As such, the total number of trainable parameters for ResNet-34 is 21,556,613 compared with 318,882 for CNN, as it is shown in Table 2. Meanwhile, this could be one reason why ResNet outperforms CNN in image classification tasks. A model with more parameters could extract more subtle features than one with less parameters.

Table 2. A comparison of number of parameters

Model	No. Parameters
CNN	318,882
ResNet	21,556,613

6. CONCLUSION

In conclusion, for both binary and multiclass image classification tasks, NN based model performs better than traditional methods like SVM and RF. It is also note-worthy that the testing accuracy for all models is lower than training accuracy, which indicates the lack of generalization capability of the model. In the next step, the research focus could be on the NN based models and create more advanced network architecture with better predicting performance. Meanwhile, the model generalization problem should be addressed, which means as long as the datasets come from the same distribution the model performance with those feeding datasets should be as close as possible.

7. REFERENCES

[1] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *IEEE transactions on Neural Networks*, vol. 10, no. 5, pp. 1055-1064, 1999.

[2] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.

[3] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, 2017: Ieee, pp. 1-6.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," presented at the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.