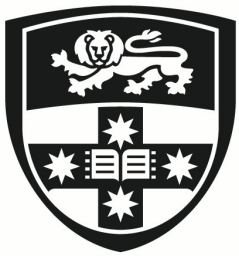


THE UNIVERSITY OF
SYDNEY

Assignment2 Group39 Report

Tutors
Daisy: Josh Nguyen
Tim: PRAC05: Mon 20:00 Tahsin Samia

Ying Dai 490066027 ydai8556 50%
Guangyi Liu 490581612 gliu9705 50%

Abstract

This report commits to applying various classification algorithms on one of the challenging datasets named cifar100 and attempt to improve the performance of the models built with the algorithms and frameworks we have and the information we collected. This attempt to implement the classification algorithm on the cigar100 data set will help us sharpen our coding skills and accumulate the experience of usage of Data preprocessing and fine-tuning in the field of data science, as well as a deeper understanding of the principles and applications of various machine learning algorithms and deep learning mining machines.

A few models including machine learning algorithms and deep learning frameworks have been successfully established and SVM+HOG, VGG, and ResNetare, the three most prominent among all the algorithms we have attempted to establish are introduced in this report. The result is that SVM+HOG only has 21% accuracy due to the uncertainy of the parameters caused by failed grid search and big amount of training data. For CNN frameworks, ResNet has the best performance (71% accuracy), which is around 11% more than that of VGG whose accuracy is around 60% and both of them perform better than the original CNN deep learning framework. From the results, traditional machine learning models require more data preprocessing and parameter adjustments to have better performance whereas the performance of ResNet on classifying cifar100 is better than that of VGG not considering all the technical and equipment limitations.

Contents

1 Introduction

Image recognition technology is a process of recognizing and detecting objects in digital images or videos. The global image recognition market has grown even more rapidly in recent years and is expected to reach $109.4 billion by 2027, according to a new report from Grand View Research.This concept is used in many fields, such as security, surveillance, medical imaging and so on. With the application of image recognition technology in various fields becoming more and more extensive, the improvement of accuracy provides him with more potential application markets, such as autonomous driving vehicles, which makes relevant research become more and more important.[11]

This project aims to build a classifier to successfully classifies the colour images of the size 32x32 of the Cifar-100(Canadian Institute for Advanced Research, 100 classes) dataset. It consists of a training set of 50,000 examples and a test set of 10,000 examples which belong to 100 different categories.

With this in mind, the project group has successfully trained and evaluated three classification algorithms, specifically a support vector machine (SVM) based on the hoga VGG and a ResNet. This ResNet has been proven to classify the dataset effectively and reached an accuracy of over 70%.

2 Previous Work

The CIFAR-100(Canadian Institute for Advanced Research, 100 classes) dataset consists of 60000 32x32 colour images in 100 classes, with 600 images per class. There are 50000 training images and 10000 test images.[12]

According to Paper with Code, there are now 3,500 literature on training this dataset, and Google Scholar shows more than 9,500 research papers cited. Cifar100 has a huge amount of data that allows deep learning researchers to quickly prototype their algorithms. A review of the current literature shows that most previous work on the Cifar100 dataset has generally been on neural networks, particularly convolutional neural networks and other forms of deep learning, because deep learning can identify nonlinear patterns in high-dimensional datasets. The Cifar100 dataset is also frequently used as an integrity check for new cutting-edge machine learning algorithms and techniques. The following table illustrates the use of Cifar100 in the current literature and online:

Author/s	Model/s	Accuracy
Pavel Izmailov,Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov. Andrew Gordon Wilson	PyramidNet-272 + SWA	84.16%
Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger	DenseNet	82.62%
Seyyed Hossein Hasanpour, Mohammad Rouhani, Mohsen Fayyaz, Mohammad Sabokrou	SimpleNetv1	78.34%

Table 1: Cifar-100 Accuracy in Literature and Online

Pyramid network is a kind of convolution network. Different from ResNet, it uses additive pyramid to gradually increase the dimensions and zero-filling directly connected identity mapping. His core idea is to increase the dimension of the feature graph gradually, rather than to increase the dimension of the feature graph sharply on each residual element by downsampling.[6]

ResNet is a deeper network layer that solves the problem of deep network gradient disappearance. However, DenseNet starts with features and achieves better effects and fewer parameters through the ultimate use of features. All layers are directly connected to ensure maximum information transmission between layers. This mitigated gradient disappearance and reduced the number of parameters to some extent.[10]

Unlike ResNet, which uses residual network, SimpleNet is a convolutional neural network with 13 layers. The network adopts homogeneous design with 3 $\times$ 3 cores for convolution layer and 2 $\times$ 2 cores for pooling operation. SimpleNet also uses batch normalization to solve the problem of gradient extinction and overfitting.[7]

3 Methods

This section mainly has thress parts: data pre-processing, classifier methods and evaluation metrics.

3.1 Data Pre-processing

Proper data preparation and preprocessing are as important as training classification models.[14] It can reduce the noise artifacts generated and provide better feature vectors which will reduce misclassification and improve the recognition rate effectively.[3]

3.1.1 Train and Validation Data Segregation

In order to train a better model, a dataset will be split into three parts: training, validation and testing sets. The training set is mainly used for fitting the model. Then we use validation set to

frequently evaluate the model which can prevent overfitting of the model.[3]

In all three classification models constructed in this project, train-valid sets and test sets are divided in an 8:2. In other words, 80% of the data is for a train-validation dataset, a train:validation split of 90:10 was performed in order to train the model and verify the performance and associated hyperparameters.The remaining 20% is used for testing.

3.1.2 Feature scaling: Z-score Normalisation

Feature scaling is a necessary pre-processing step to improve the validity of the model due to the differences in the size of variables in the dataset.[3] The most common methods are min-max Normalization and Z-score normalization. Normalization maps data uniformly to the [0,1] interval while standardization turned the data into a distribution with a mean of 0 and a standard deviation of 1.[1] After studying the data, we decided to use standardization.

$$x' = \frac{x - \mu}{\sigma}$$

Fig. 1. Z-score Normalisation formula [1]

3.1.3 Feature extraction: Hog

Histogram of oriented gradient (HOG) is an efficient feature processing method. Its feature is a feature descriptor used for object detection in computer vision and image processing, usually used in portrait recognition in conjunction with SVM.

Hog first grays the image, then standardize the color space of the input image with gamma correction method, adjusting the contrast of the image and reducing the impact caused by local shadow and illumination change of the image, and suppresses the interference of noise. Then, it calculates the gradient of each pixel of the image, captures the contour information, and further weakens the interference of illumination, Next, divides the image into small cells, counts the gradient histogram of each cell to form the descriptor of each cell, then forms a block of every few cells, connects the descriptors of all cells in the block in series to become the hog feature descriptor, and then connects the hog feature descriptors of all blocks in the image to obtain the feature vector, which will be delivered to the model establishment.

Before the training process of SVM, we divided the image into 8x8 four pixel small units. Through this division, we reduced the dimension of training data, sped up the training model, and improved the performance of the model by obtaining effective features.

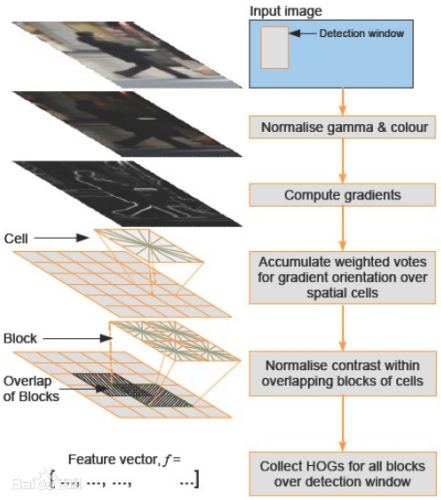


Fig. 2. Hog Principle

3.1.4 Data Augmentation

Deep learning is highly data-dependent, so training a good model requires enough data to support it. When there is too little data, it is easy to overfit. Therefore, we can use methods such as data enhancement to increase the amount of data.

From the neural network's point of view, photos taken from different angles are several different, unique objects. Therefore, in order to obtain more data, we only need to make some minor modifications to the existing data set, such as flipping, panning, or rotating. Our neural network thinks these are different images.[5]

In this project, we use ImageDataGenerator in Keras to enhance the data, which can enhance the sample data in each batch, expand the size of the data set, and enhance the generalization ability of the model, such as rotation, deformation, normalization and so on.

3.1.5 One-hot encode

One-hot coding is a way to turn data into algorithms and get better predictions. The method is to use an n-bit state register to encode N states, each of which has its own register bits, and only one of them is valid at any time. It maps the categorical value to integer values and then, represented as a binary vector, which is zero except for the index of the integer, which is marked with 1. It solves the problem that the classifier can not deal with discrete data.[2]

In the cifar-100 dataset, there are 100 classes which are categorical variables. Because many machine learning algorithms cannot operate directly on label data, thus we convert labels to numeric by one-hot encoding.

3.2 Classification Models

In this section, we will outline the models: SVM+HOG, VGG, ResNet

3.2.1 Support Vector Machine (SVM)

SVM is a well known supervised learning algorithm. It is usually used for the classification of high-dimensional data and often has good performance. There are a large number of kernel functions that enable it to participate in non-linear classification problems and can be well normalized unless the data is massive.

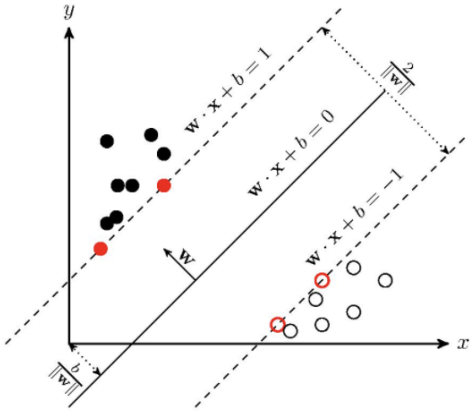


Fig. 3. SVM Principle

From the figure above, the priciple of SVM is decided by the kernel funtion and the value of C. Grid Search and Hog were planned to be used in the establishment of SVM model since it helps us to find the best value of C and the best kernel function for this dataset, which justify the principle how the SVM will clarify the data into different groups.

Since SVM+HOG did well in the classification task of assignment one and it is wildly used to detect pedestrian and other images, we chose SVM+HOG as one of our models.

3.2.2 Convolutional Neural Networks (CNN)

Convolutional neural network (CNN) is a deep learning algorithm that takes input images, assigns importance (learnable weights and biases) to various aspects/objects in the image, and is able to distinguish between them. His structure resembles the pattern of connections between neurons in the human brain and is inspired by the organization of the visual cortex. Individual neurons respond to stimuli in only a limited region of the optic field, the receptive area. The collection of these areas overlaps and covers the entire visual area.[15]
Figure 1 shows VGG16(one of the methods used in this project), a 5-hidden layer CNN architecture. In addition to the initial layer and output layer, it contains five hidden layers: Convolution operator,

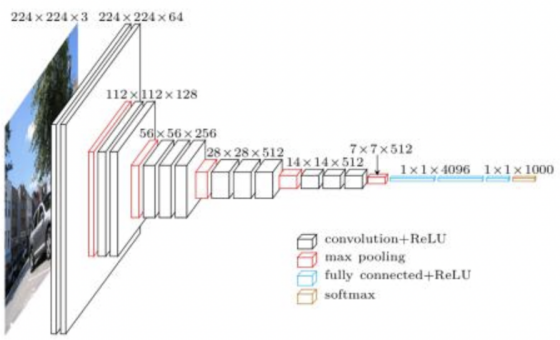


Fig. 4. CNN Architecture

Rectified Linear Units (ReLU), Max-pooling operator, Batch Normalisation's and Dropout operator.

3.2.3 Batch Normalisation

In the training of deep neural network, the change of the parameters of the previous layer will affect the input of the next layer. Batch normalization solves the problem of internal covariable displacement by normalizing each small Batch while training the network itself, which means that a higher learning rate can be used.[18]

However, if each layer is normalized, some of the weight changes made by the previous layer and noise between the data will be lost, which may result in suboptimal weights being passed on. As a result, batch normalization adds two trainable parameters, γ and β, that can scale and move normalized values.[18]

$$\hat{x} = \frac{x - E[x]}{\sqrt{Var[x]}} \qquad y = \gamma \cdot \hat{x} + \beta$$

(a) Normalised Vector (b) Shift Normalised Value

Fig. 5. Batch normalised formula

3.2.4 Activation Function: Rectified Linear Unit (ReLU)

Activation functions convert input into a range of outputs and can help networks learn complex patterns in data. Without the activation function, the output signal becomes a simple linear function.[13]

ReLU functions are simple and do not involve heavy computation because there are no complex mathematical operations. Therefore, the model can take less time to train or run, reducing the

impact of the vanishing gradient problem.[13] Following is the formula:

$$\mathbf{f(x)=max(0,x)}$$

The function returns 0 if the input value is negative, or x if the input value is positive. Therefore, it gives an output range from 0 to infinity.[13]

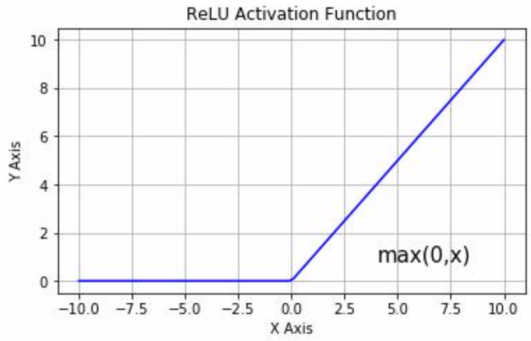


Fig. 6. ReLU Activation Function

3.2.5 Dropout Function

In machine learning models, if the model has too many parameters and too few training samples, the trained model is easy to produce over-fitting phenomenon. That is, the loss function of the model is small in training data and the prediction accuracy is high. However, the loss function of the test data is relatively large and the prediction accuracy is low. Dropout can effectively mitigate the occurrence of overfitting and achieve regularization to some extent.[9]

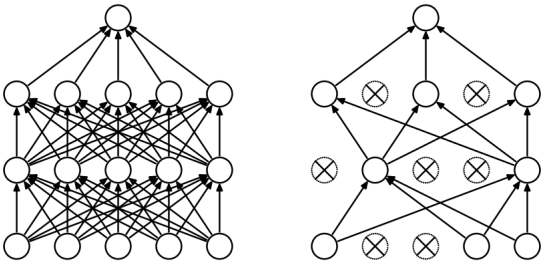


Fig. 7. Diagrammatic Respresentation of Droupout in a Standard Neural Network.[**?**]

3.2.6 Final Classification: Softmax

The Softmax function translates the predicted results of the model into an exponential function. The input values can be positive, negative, or zero, and softMax converts them to values between 0 and 1 so that they can be interpreted as probabilities. If the input value is small or negative, SoftMax converts it to low probability, and if the input value is large, it is high probability, but it will always remain between 0 and 1.[20]

$$softmax(z_i) = \frac{exp(z_i)}{\sum_j exp(z_j)}$$

Fig. 8. Softmax Formula [16]

It is applied to the final layer of the network and of the 100 possible classes to calculate.

3.2.7 VGG

VGG builds depth models by reusing simple base blocks. Its convolutional layer uses a very small receiver field (3x3). There is also a 1x1 convolution filter, which maintains spatial resolution, as a linear transformation of the input, followed by a ReLU element.[19]

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig. 9. VGG configurations[17]

VGG uses deeper network structure, smaller convolution kernel and pooled sampling domain, which enables it to control the number of parameters while obtaining more image features, avoiding too

much computation and too complex structure. Thus we choose VGG16 as one of our models.

3.2.8 ResNet

Resnet(Deep residual network) was introduced in 2015, a milestone in the history of CNN imagery. A residual learning framework is proposed to alleviate network training, accuracy can be obtained from greatly increased depth.[8]

Theoretically, when the number of network layers increases, the network can perform better feature extraction. However, the depth model has problems such as gradient disappearance, so sometimes the deeper the network, the lower the accuracy. Residual learning can improve this problem on a certain basis.

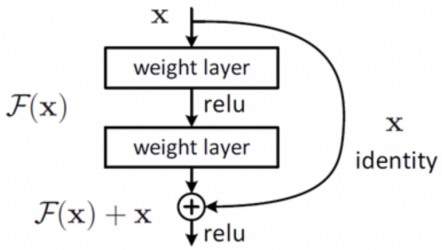


Fig. 10. Residual learning block[8]

The structure of residual learning is shown in Figure 8. It is a shortcut connection because it is similar to a short circuit in a circuit. When the input is x, the feature it learns is denoted as H(x). In ResNet, it learns the residual F(x) = H(x) -x, making the original learning feature F(x) + x. The reason for this is that residual learning is easier than raw feature learning. Network performance does not degrade when the residual is 0 because the accumulation layer only does identity mapping. The residuals will not be zero, enabling the accumulation layer to learn new features based on the input features, resulting in better performance.[8]

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 11. ResNet configurations[8]

ResNet increases network depth while avoiding negative results. So we can add depth, but we have

faster training and more accuracy.[4] Thus, we choose ResNet18 as one of our models.

4 Experiments result and discussion

4.1 Accuracy

4.1.1 SVM

The first model we applied on cifar100 dataset is Support Vector Machine(SVM). The reason why we chose SVM+HOG is that this is usually used for pedestrian classification and it worked well for the classification last assignment. The scenario was supposed to be that after the Hog feature processing, grid search would be used to find the best parameters that helps SVM perform best. However, the first difficulty we met is that the datacet of cifar100 is relative huge, which reduced the accuracy of SVM+HOG. When we run grid search on SVM, it took too long to have a result.We also had a thought to stop using HOG, but the dimontion of the data is too high to pass in the SVM model. Thus, these nice tools for SVM does not work well for cifar100 classification. Finally, I decided to caluculate the accuracy of SVM when C equals to 1,5 and ten and it turns out the accuracy does not have huge changes.

Thus, based on the situation we had, we chose to set C as default, which makes the accuracy of final SVM stable at 21%.

4.1.2 VGG

According to the literature Very Deep Convolutional Networks For Large-scale Image Recognition[17], our team implemented the VGG network structure from scratch.

Vgg16 has 16 layers (excluding Max pooling layer and SoftMax layer). All convolution cores use the size of 3x3, and max pooling uses the size of 2x2. Also, I have added a layer of Batch Normalization and Dropout behind each layer to speed up model convergence and prevent overfitting.

In the beginning, We constructed a 16-layer network structure of 2x64+3x128+3x256+3x512+3x512 according to the structure in the literature. However, we adjusted the learning rate, optimizer, dropout, and other parameters, but the results were poor, the accuracy rate was only 60% after running 100epoch. So, based on that, we modified the network structure to make it 2x32+2x64+3x128+3x256. In this structure, it only needs to run 60 epochs to get to 60%.

4.1.3 ResNet

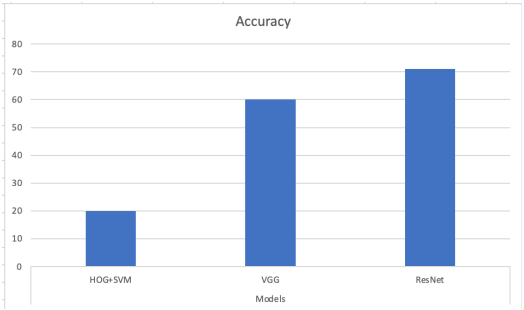
According to the literature Deep Residual Learning for Image Recognition[8], our team implemented the ResNet18 network structure from scratch. So 18 is 1+(2x2 +2x2 +2x2 +2x2) +1, where the first 1 and the last 1 represent an ordinary convolution layer at the beginning and the last fully connected layer (Relu, BN and other layers without weight are not counted here). The four in the middle are four ResNet layers, each layer contains two blocks, and each Block has two convolution layers. The initial input format of the basic ResNet structure is 224x224. Since the image size of CIFAR-100 is 32x32, I changed the kernel size of the first layer from 7 to 3.

At the beginning, we trained the model with the fit function of TensorFlow, and the accuracy was up to 50%. Later, we decided to write the training and test function by ourselves. We chose gradient tape to record a series of operations performed on some inputs and produce some outputs so that we could differentiate the outputs based on the inputs and perform gradient descent optimization. Finally, the accuracy rate reached 71 percent at 60 epoch.

4.2 Discussion

Figure 12 allows us to compare the accuracy of three classifiers SVM, VGG and ResNet. In addition to average accuracy, the following additional evaluation indicators were identified for each model include:[3]

- Training time - the time to train the model.
- Recall - the ratio of correctly predicted positive observations to the all observations.
- Precision - the ratio of correctly predicted positive observations to the total predicted positive observations.
- f1-score - the weighted average of precision and recall.[3]



(a) Accuracy Comparison

Model	Parameters	Accuracy-Test	Precision	Recall	F1-score
SVM	C = 1	20.96%	18%	21%	17%
VGG	4 hidden layers, 32 filter, stride = 1, padding=" same" & maxpool(2,2)applied. Batch-size = 128	57.10%	61%	57%	57.08%
ResNet	4 residual block, 64 filter, 1 GlobalAveragePool	70.7%	71%	71%	70.6%

(b) Overall Performance Index

Fig. 12. Summary of Performance

From figure 12(a) and the evaluation of the results above, it is immediately apparent that the ResNet model shows the best accuracy results of over 70% among them.

From figure 12(b), compared to VGG and ResNet, the performance of SVM model is not very skilled.

Although SVM, as a machine learning, has a faster training speed compared with the two deep learning models VGG and ResNet, it cannot construct an effective linear classifier and can not classify the multi-class tags in cifar100 data set well.

The results of this report show that although the performance of support vector machine itself is good, it is surpassed by the deep learning algorithm in the ciFAR100 data set. Deep learning

algorithms are more suitable for high-dimensional image classification than he is. Support vector machines are inherently unsuitable for multi-class problems, which means that the distance measures used in support vector machines need to be pruned.

4.3 Critical Reflection

4.3.1 Daisy

Before this project, I only knew about traditional CNN, but not about VGG, Resnet and other model frameworks. In the process of learning, I learned that stacking different layers and how many layers can have a big impact on the result. If I have extra time, I will learn how to build more complex machine learning models and explore other in-depth models besides CNN.

In the process of training the model, I still didn't know how to choose the parameters such as learning rate and optimizer due to my lack of experience. After this learning, I successfully realized the model and accuracy that I thought could not be realized earlier. Although the mission was a challenge, it was also an opportunity to learn and improve.

4.3.2 Tim

In order to complete this project, I have checked many materials of CNN architecture models, such as Alex net, VGG, GoogLeNet, etc., and learned from the source code, instructional videos on the Internet. In the course of this work, I attempted to build google net and Alex net to predict the cifar100, but the results were not ideal, and because of limitation of time and personal ability, these are not specifically reflected in the report. In the process of cooperation, we used the pair-programming strategy to maintain high-frequency communication. My time should be better arranged, perhaps more models can be made.

5 Conclusion and future work

From the perspective of the construction process and overall performance of the SVM+HOG model on cifar100, there is a significant gap between traditional machine learning algorithms and the neural network framework. It is clear that the performance of SVM regarding accuracy and time consumption has been greatly reduced when the amount of data reached a massive amount. Its accuracy is similar to the accuracy of the original CNN framework before adjusting any parameters. Instead, ResNet and VGG on cifar100 perform significantly better than the SVM+HOG framework and also the original CNN framework, which is due to the fact that effective layer structures or deeper depths are embedded in them compared to the original version. And in a large-volume data environment, deep learning frameworks have reliable performance and demonstrate their advantages. With continuous iteration and training, its accuracy has been increasingly improved in different paces although what deserves attention is to prevent over-fitting and large time-consuming.

In order to improve the shortcomings of our current models, more research can be conducted to improve the efficiency of conducting the code and reduce the time consumption. Besides, more advanced methods to extract features from images or compress images can be explored to improve the performance of the models whose accuracy is not satisfactory. Finally, some more CNN frameworks such as AlexNet and GoogLeNet can be attempted to compare various CNN frameworks'

performance together.

6 Appendix

6.1 Loading the file

Upload the files to colab, then run.

6.2 Running each model

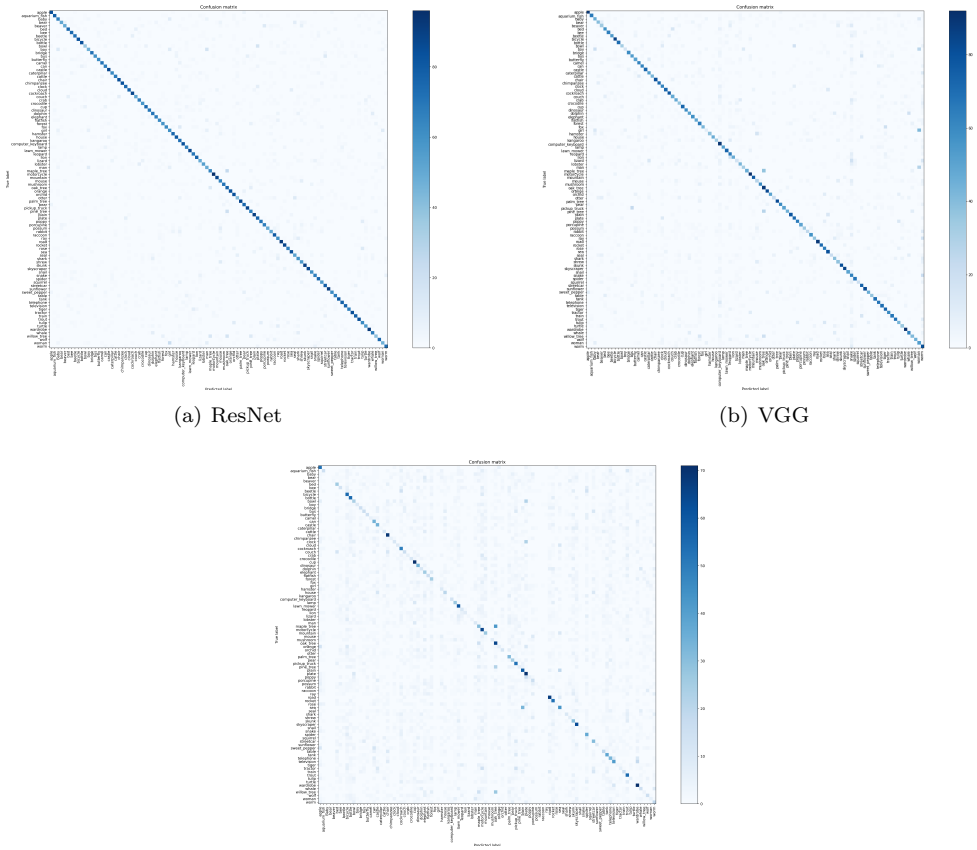
SVM
Running "Group39SVMalgorithm.ipynb"
Needed Library: tensorflow, sklearn, keras, skimage, matplotlib.pyplot, time, numpy

VGG
Running "Group39VGGalgorithm.ipynb"
Needed Library: tensorflow, keras, matplotlib.pyplot, time, numpy
ResNet

Running "Group39bestalgorithm.ipynb"
Needed Library: tensorflow,sklearn, keras, matplotlib.pyplot, time, numpy
ResNet

6.3 Confusion matrix

For high quality confusion metrics, you can see it in the ipynb files or in this link "https://drive.google.com/drive/folders/
$vFKut2p?usp = sharing$"



(a) ResNet

(b) VGG

(c) SVM

Fig. 13. Summary of Confusion Metrics

References

[1] Zaid Alissa Almaliki. Standardization vs normalization, 10 2020.

[2] Jason Browniee. Why one-hot encode data in machine learning?, 07 2017.

[3] Daisy Dai and Time Liu. Comp5318: Assignment 1, 2021.

[4] Baki Er. Microsoft presents : Deep residual networks, 08 2016.

[5] Arun Gandhi. Data augmentation — how to use deep learning when you have limited data, 11 2018.

[6] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks, 2017.

[7] Seyyed Hossein Hasanpour, Mohammad Rouhani, Mohsen Fayyaz, and Mohammad Sabokrou. Lets keep it simple, using simple architectures to outperform deeper and more complex architectures, 2018.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[9] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors, 2012.

[10] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[11] Grand View Research Inc. Image recognition market size worth $109.4 billion by 2027 — cagr: 18.8%: Grand view research, inc., 03 2020.

[12] Alex Krizhevsky. Cifar-10 and cifar-100 datasets, 2009.

[13] Hussain Mujtaba. An introduction to rectified linear unit (relu) — what is relu?, 08 2020.

[14] Srishti Saha. Data preparation and preprocessing is just as important creating the actual model in data sciences. . . , 10 2018.

[15] Sumit Saha. A comprehensive guide to convolutional neural networks—the eli5 way, 12 2018.

[16] Shipra Saxena. Softmax — what is softmax activation function — introduction to softmax, 04 2021.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[18] Robin Vinod. Batch normalisation explained, 05 2020.

[19] Jerry Wei. Vgg neural networks: The next step after alexnet, 07 2019.

[20] Thomas Wood. Softmax layer, 05 2019.