Advanced Machine Learning report

Haoen Feng
University of Southampton
hf1y18@soton.ac.uk

Yinyu Jin
University of Southampton
yj7n18@soton.ac.uk

Yibo Liu
University of Southampton
yl5y18@soton.ac.uk

Xiaoyu Wang
University of Southampton
xw1g18@soton.ac.uk

ABSTRACT

This report focuses on the procedures we explored in the tissue cancer detection competition. Some of approaches boost the accuracy and improve the generalization while others are found not suitable for this project. The coefficients and option settings we choose in the final version of prediction model are all determined after many experiments of testing and verification in order to reach a best performance.

KEYWORDS

cancer detection, neural network, training procedure refinements

1 INTRODUCTION

Due to their outstanding accuracy and robustness, convolution-based neural networks have become a dominant method for image classification. The steady improvement of classification accuracy has also let us witness their potential and promising applications. For example, the AmoebaNet[11] model proposed in 2018 has achieved top-1 accuracy of 84.3% on ImageNet, which shows a significant improvement compared to the AlexNet (top-1 accuracy of 62.5% in 2012[6]). Besides the improved models, refinements of training procedures, such as data pre-processing, output post-processing, and the changes in optimization and loss function also contribute a lot in the accuracy improvement with theoretical support. These training techniques can be combined together to further improve the performance of the neural network, which is significant in the Kaggle competition, where a small difference in accuracy can make a huge gap in the leaderboard. After applying several training procedure refinements, our model finally achieves 98.15% accuracy from the baseline model (97.20% accuracy), which ranks number 16 in the final leaderboard. The following experiments will demonstrate details of our neural network design and reasonable selections of training procedure refinements in the cancer recognition problem.

2 PREPARE PROCESS

2.1 Dataset information

Format	size	channels	Bits per channel
.tif	96x96	3	8
Training set	Validation set	Data type	Class number
220K	57K	Unsigned char	2

2.2 Principal components analysis

2.2.1 the objective of PCA. When dealing with a large dataset, reconstructing the dimensions of the dataset is an efficient way to accelerate the training. Principal components analysis (PCA) is a commonly used dimension reduction method that preserves the most important parts of high-dimensional data and removes noise or unimportant features. In our experiments, we tried to pre-process our data using PCA to explore whether it does work in this problem.

The essence of PCA is to find some projection directions so that the data has the largest variance in these projection directions (these projection directions are orthogonal to each other). We need to calculate the variance of the original data projected on these orthogonal bases. The larger the variance is, the more information is contained on the corresponding orthogonal basis. Therefore, The eigenvectors with the large eigenvalues are the projection directions with the corresponding large amount of information. In the implementation, we can obtain eigenvalues and eigenvectors using singular value decomposition (SVD).

2.2.2 the implementation of PCA. In the beginning, we tried to implement it using the traditional PCA method. First, we convert the images of the dataset into a $m \times n$-dimensional matrix, where m is the number of the images, and the n represents pixel-based flattened data. Each element in flattened data subtract their average and then compute the covariance matrix. With the use of SVD, the eigenvalues and eigenvectors of the covariance matrix are calculated so as to extract the eigenvectors corresponding to the top several eigenvalues. The percentage of PCA represents the proportion of the variance. Our experiments use different percentages of dimensionally reduced datasets from 10% to 90% to explore their performance. However, the pixel values of all images are calculated in a matrix, which leads to huge computational time.

In our implementation, due to the large size of the original dataset, an error occurred while calculating the matrix covariance using the np.linalg.eig() function. Thus, we used SVD sub-channel instead in the PCA dimensionality reduction. We split the image by the number of channels, and apply SVD on the pixel matrix of each channel separately using the np.linalg.svd() function which returns the left singular matrix u, diagonal matrix σ, and the right singular matrix v. After that, we can extract the top k values from those 3 matrixes so that reconstruct the reduced-dimensional image matrix using the formula $a = u\sigma v$. Finally, the results from all channels are connected together to obtain the integrated reduced-dimensional image matrix.

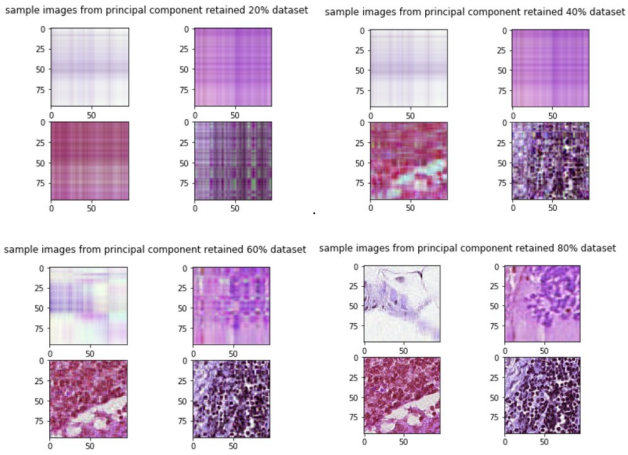

Figure 1: the images after PCA dimensionally reduction with different percentages

The figures show the results of partial dimensionality reduction in descending order of principal component retention. It can be seen that the less the main component is retained, the more blurred the pictures are. When the main component is kept above 60%, the picture becomes more clear.

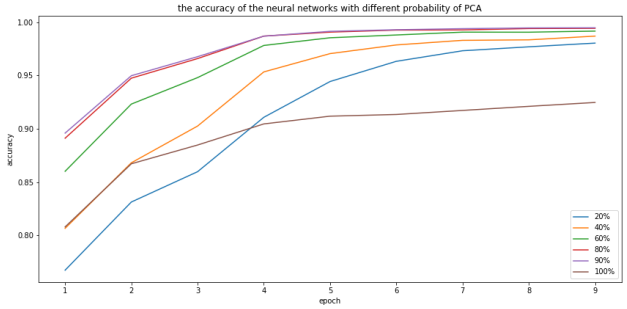

Figure 2: the accuracy of the neural networks with different probabilities of PCA

The figure shows that different probabilities of components in Principal components analysis are still able to make the training converge to a high train-set accuracy. When using PCA, the larger the probability of principal components are used, the higher the accuracy we can reach, but the difference becomes small when the probability is above 80%. However, the performance of training without PCA is worst among them, which means PCA did extract meaningful information for the training.

2.3 Data augmentation

There are numbers of ways to avoid overfitting; more data, early stopping, augmentation, regularization ,and less complex model architectures. In this situation, the dataset is provided by the organization, so it is fixed. At this stage, we focus on augmentation which is a very practical and efficient way of making the best use of given data. Data augmentation is to reduce manual interventions to extract salient information. Due to the fact that the object location and shape are not clear in each image, extracting meaningful information from the images in advance would significantly enhance the

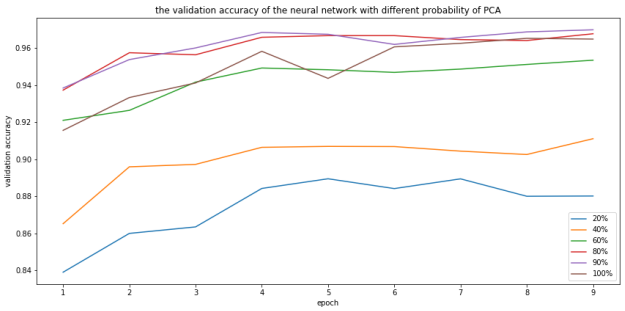

Figure 3: the validation accuracy of the neural networks with different probabilities of PCA

The figure shows that although the training accuracy is increasing with the number of epoch rising, the validation accuracy shows different trends. The smaller probability of PCA converges at a lower level of validation accuracy. Therefore, small principle components analysis is not of benefit to the generalization of the model. With respect to the PCA with a probability above 80%, the trend of validation accuracy is similar to or even better than that without PCA. The 90% PCA reach the highest level of validation accuracy.

image quality and boost the classifier performance. Several useful libraries like albumentations and imgaug help us simplify the implementation of various transformations into the images of dataset. Since that we receive little information about the cancer tissue size and shape, the data augmentation we set focuses on the transformations including cutting, rotation, contrast, color, brightness and so on in order to make the model learn how to locate the cancer tissue automatically. Different transformations have different weight for the classification as some of them may introduce noise into the images. Thus, we set different frequencies and priorities to different transformations to reduce the risk of noise, as well as make sure enough exploration. Here is the detail of data augmentation using albumentations library:

1. Resize the image into a 288-by-288 square shape and then normalize RGB channels with the mean (123.68, 116.779, 103.939) and variance (58.393, 57.12, 57.375) computed on ImageNet.

2. Flip horizontally or vertically with 0.5 probability.

3. Flip horizontally or vertically and then rotate 90 degrees with 0.5 probability.

4. Randomly choose one of transformation including brightness, hue, saturation, contrast and different combinations of them with coefficients uniformly initialized as 0.1, 0.3 and 0.5.

5. Use the Gaussian filtering and Average filtering, and then sharpen and emboss the image with 0.2 probability.

We also implement the same augmentations during validation so as to better observe the performance of generalization. From the comparison of figure 3 and figure 4, it can be seen that using rotation and flipping improve the accuracy a lot from the baseline without augmentation, but the color-processing perform poorly in this problem, which is probably because some over-processing destroy the feature of cancer tissue. Thus, we only apply the rotation and flipping in this part.

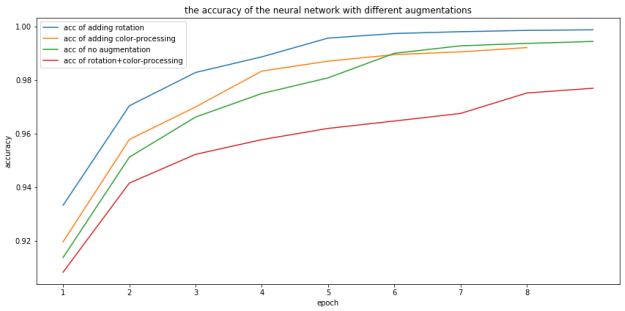

Figure 4: the accuracy of the neural network with different augmentations

The figure shows that the performances of different data augmentations, where the rotation represents the flipping and different angles of rotation operation for images. The accuracy of data-set with rotation augmentation rises more significantly and converges at the highest level, compared with that from other operations. Either color-processing or rotations are beneficial to accelerate the convergence, but the combination of them cannot boost the performance of the training.

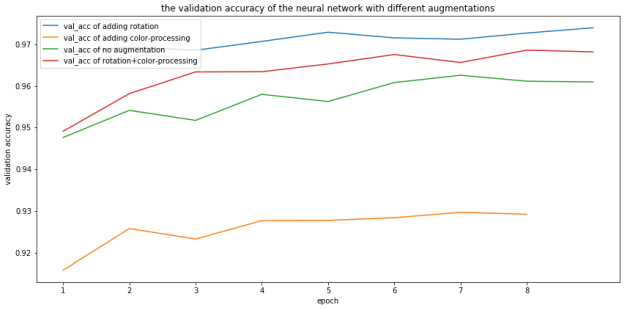

Figure 5: the validation accuracy of the neural network with different augmentations

The figure shows that adding the rotation and flipping augmentation can also boost the validation accuracy and reach the best performance than others, and thus we decided to apply this augmentation in our training. Although the training without augmentation performs well in the training accuracy, its validation accuracy shows it has a tendency of over-fitting. The operation of color-processing performs poorest among all.

3 NEURAL NETWORK CLASSIFICATION

3.1 Convolutional base

As we need to capture the tissue feature which is different from any object in ImageNet or Coco dataset, using the pre-trained weight directly on a different dataset would result in under-fitting. Although retraining an advanced neural network from scratch would better learn specific feature for the tissue images, expensive computational resource and unstable convergence are still the obstacles to achieve a good performance. Since the weights of low layers have the ability to extract basic feature such as lines and corners, fine-tuning the pre-trained model is an efficient way to adjust the upper layer weights to focus on the cancer detection. Considering the

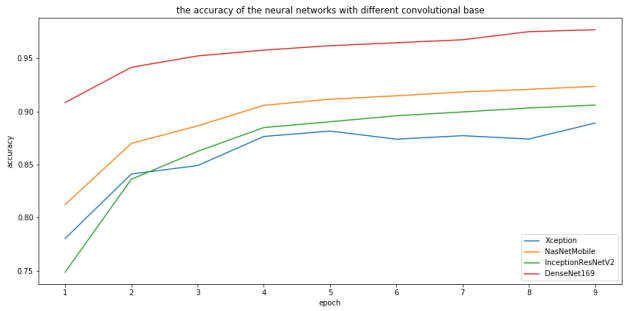

Figure 6: the accuracy of the neural networks with different convolutional base

The figure shows that all models start to converge at about 4 epochs, but DenseNet169 can achieve significantly better accuracy (95.8%) than others, followed by NasNetMobile (91%), InceptionResNetV2 (90%), and Xception (88%).

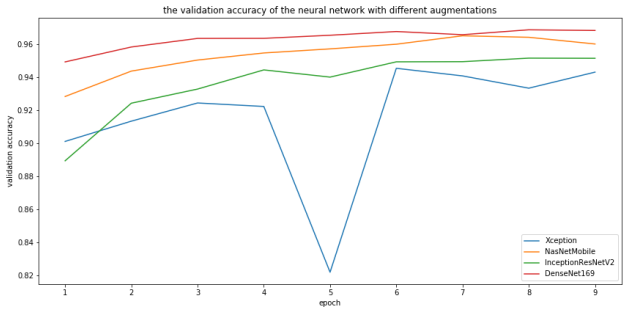

Figure 7: the validation accuracy of the neural network with different convolutional base

The figure shows that the validation accuracy of DenseNet169 is in lead throughout the whole training process. DenseNet169, InceptionResNetV2, and Xception have stable tendencies to rise, but Xception has a huge fluctuation from epoch 4 to 6, even though the training-set accuracy is not influenced. Therefore, we choose DenseNet169 as the convolutional base of our neural network.

performance of various neural networks and their model size, NasNet, Xception[2], InceptionResNetV2[13], and DenseNet169[5] are chosen as the candidates for the backbone of our model. DenseNet is a powerful convolutional model which can integrate low-level and high-level features together by densely connected network so that it would perform much better than the original convolutional network. We should note that using too deep model architecture would cause overfitting. The experiments show that all of them can achieve above 97% classification accuracy and thus the DenseNet169 is selected as the backbone.

3.2 Classifier selection

3.2.1 Convolutional neural network. The classifier layers on the top of the neural network are designed to evaluate and classify the information from the convolutional base. After the average-pooling layer extracting the feature from the convolutional base,

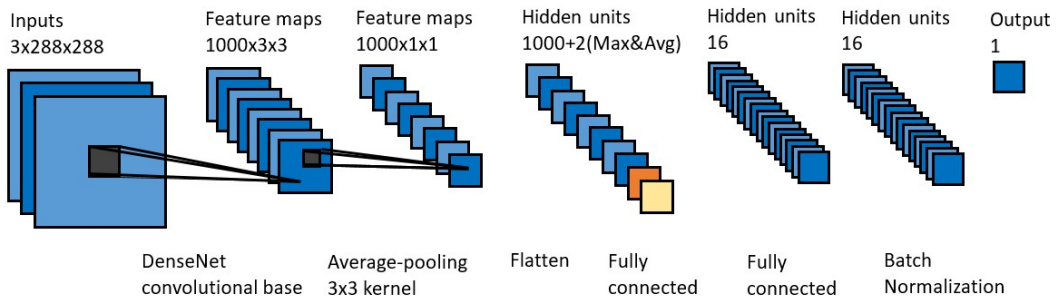

Figure 8: The final version of our cancer classification model

The figure shows that we choose DenseNet169 as the feature extractor and use a combination of few fully connected layers and batch-normalization layers as the neural network classifier.

we can get a 1000x1 dimension feature containing abstract information. To further distill the data, the maximum unit and the average are concatenated to the original data into a 1002x1 dimension feature. The experiments shows that such extension increases over 5% of classification accuracy. In addition, dropout layers and batch-normalization layers are also added into the classifier structure to fight the overfitting and accelerate the training. Relu activation is used after the Batch-normalization layers since the original Batch-normalization paper indicates that it is more likely to produce a stable distribution of data for the activation. Although many conference papers argue that the opposite way is more reasonable with beautiful theories behind them, it performs worse in our cancer problem. The vectors γ and β in Batch-normalization layers are initialized to 1 and 0 respectively. In the final output layer, it is designed as a fully-connection layer with one output and sigmoid activation. The sigmoid function can restrict the output into a range from 0 to 1, which represents the probability of having the cancer tissue in the input image. The weights of fully-connected layers are initialized with the Xavier algorithm.[4] In order to avoid large gradient updates ruining the pre-trained weights, the weights of the classifier layers should be properly trained before all the parameters in the neural network being fine-tuned.

Once we have decided which model to use, the next step is to determine the optimal hyperparameters such as learning rate, weight decay momentum, etc. Although many excellent models have been developed during recent years, the decision of hyperparameters is still remaining as a black art that needs people's experience and continues try. A grid search or random search [1] is computationally expensive and time-consuming. Furthermore, the performance is highly dependent on the human's choice. Here, we find Leslie's approach [7]: Learning rate is one of the most crucial hyper-parameter in the deep learning neural network. Once the learning rate is too big, the model will learn fast at the beginning, but it may pass over the minima since the step is too big. On the other hand, if the learning rate is too small, the model would take plenty of time to get to the minima or worse, never get to our destination. Therefore, it is important for us to determine a proper learning rate which is not too big nor too small. In the experiment, We set the maximum and minimum learning rate as the upper and lower bound of one cycle. The lower bound is usually a tenth of the maximum learning rate. We run learning rate finder in FastAI library and determine the maximum learning rate. We would better choose a value that is still aggressive (so that we train quickly) but still on the safe region from the explosion. Hence, in this case, the learning rate in one cycle should increase linearly from 2e-03 to 2e-02 in the first half cycle and decrease linearly in the second half cycle. It is also a good idea we decrease the learning rate further to a very small value to explore the minima in the end.

Weight decay is the L2 penalty of the optimizer to avoid overfitting. Leslie suggests running the learning rate finder with several values of weight decay, and then selecting the largest one which still can make us use a high learning rate. Hence, we do a small grid search with 1e-2, 1e-3 and 1e-4 decays. As shown in the figure, we find the orange line achieves the lowest loss and the minima refers to a higher learning rate. Hence it is a good idea to choose 1e-3 as our weight decay value. We also find out that for different weight decays, the learning rate corresponding to the minimum loss is slightly different. It is important the way we tune all the other hyperparameter will impact the best learning rate. Therefore, we should adjust our previous learning rate if it is necessary.

3.2.2 Convolutional support vector machine. Due to the fact that the logistic loss does not go to zero even if the data is correctly classified, it would lead to fluctuation or degradation in the accuracy. The support vector machine does not meet this problem since it just updates the parameters using the support vector data while ignoring the data far away from the hyper-plane. Therefore, SVM is another reasonable choice of classifier.

The objective of the support vector machine is to find a hyper-plane with the maximum margin that distinctly classifies the data points. For some complex problem where the data distribution is non-linear, The kernel tricks are proposed to map the original dataset into a higher dimension space for a better linear classification. Inspired by Tang who proposes that simply replacing softmax with linear SVMs brings a significant improvement on datasets MNIST and CIFAR-10, we use Gaussian radial basis function as the non-linear kernel in SVM to classify the unknown data output from the convolutional base of DenseNet169. However, the validation accuracy we get from the SVM (90.12%) is much lower than that from the neural network (above 97%). Salazar et al. have compared the performance of SVM and logistic regression by statistical

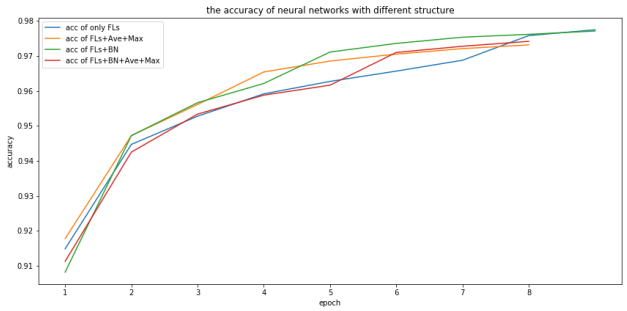

Figure 9: the accuracy of neural network with different structure

The figure shows the overall pictures of different neural network structure have a similar tendency in the training process, and all of them can make the accuracy reach above 97.5% with the same convolutional base. FLs represents that the final classifier is made by 2 fully-connected layers. Max and Ave stand for the maximum parameter and the average value, respectively. BN is the Batch-normalization layers for short.

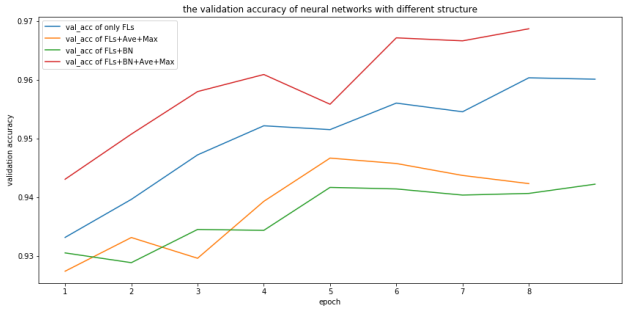

Figure 10: the validation accuracy of neural network with different structure

The figure shows that four neural network structures have different performance compared with that in the training accuracy. Among them, the combination of fully-connected layers, batch-normalization layers, averaging and maximizing operation reach the highest validation accuracy. Thus, we choose this structure for the final version of the neural network.

simulation.[3] They found that logistic regression performs better than SVM When the sample data is unbalanced. For the Poisson, Exponential and Normal distributions, SVM is superior to logistic regression. Therefore, the unbalanced cancer dataset is a possible reason for the bad performance. Also, since the back-propagation cannot be applied to the SVM for fine-tuning the parameters in the convolutional base, the pre-trained weight we use in the convolutional base is the result of the convolutional neural network. This may lead to a limitation in the SVM classification.

3.3 Loss function

3.3.1 Triplet loss. Triplet loss is our first selected loss function due to its outstanding performance in classification among classes with high similarity. FaceNet is the best example of using the triplet loss to 'learn a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face

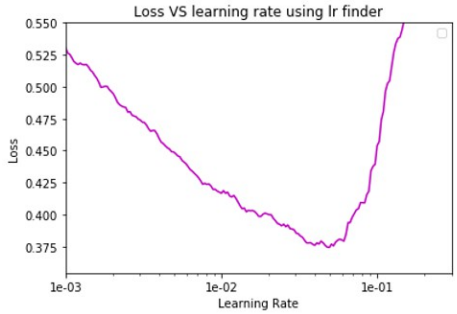

Figure 11: the change of loss with different learning rate

The loss decreases as the learning rate starts with a very small value. Then it stopped decreasing and begins rapidly diverse from the point of 3e-02. It is a risk using just the bottom point, since the model may diverge. A bit earlier than the bottom like 2e-02 should be good value to choose as the maximum learning rate.

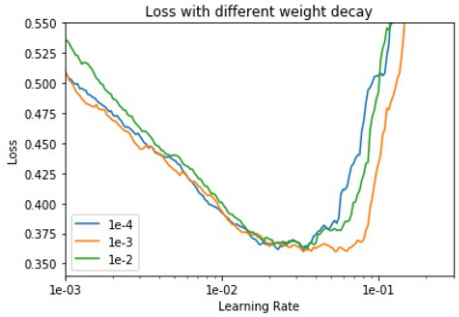

Figure 12: the change of loss with different weight decays

The figure shows that corresponding learning rate bottom is different when the different weight decays are applied.

similarity'.[12] In other words, triplet loss manages to make the face recognition into a distance-based classification like K nearest neighbor, which is also significant and useful in our cancer problem. The purpose of triplet loss is that

$$||f(x_i^a) - f(x_i^p)||_2^2 + \alpha < ||f(x_i^a) - f(x_i^n)||_2^2 \quad (1)$$

$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \tau \quad (2)$$

where α is a fixed margin limitation between negative and positive pairs. τ is the combination of all possible triplets reconstructed from the original dataset and the number of it is N. The final loss function is

$$L = \sum_i^N [||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha]_+ \quad (3)$$

With respect to the cancer classification, the model is trained to make the images with tumor tissue have small distances and the images with normal tissue have large distances in the embedding space. The hyper-parameters margin can be adjusted to meet different situation corresponding to class similarity. For accuracy improvement, hard-example mining techniques are also designed to automatically

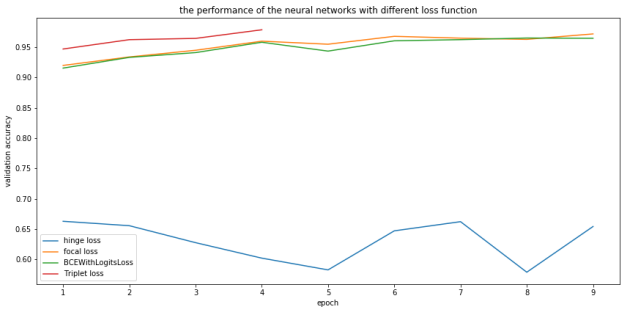

Figure 13: the performance of the neural networks with different loss function

The figure shows that different loss function leads the training to different learning directions. The focal loss performs similarly to the cross-entropy loss, but hinge loss does not work in this problem. Triplet loss reaches the highest validation accuracy among the training process, but the time cost in each epoch is 3 times as larger as others due to the fact that it has to make a comparison of the anchor image, the positive image, and the negative image for each update.

selecting the difficult samples for retraining, which is beneficial to faster convergence and higher efficiency.

3.3.2 Focal loss. A balanced dataset indicates that each class is evenly distributed with a similar number of samples, which contributes to learning various useful signals from different classes. If one class is overrepresented in the training dataset, it would overwhelm the training leading to poor performance in prediction. Therefore, in the cancer dataset where nearly 60% of images are labeled as the negative samples containing the normal tissue, the negative label is more likely to dominate the total loss resulting in local minima during the training process. To address the class imbalance, focal loss is first introduced in the one-stage object detection to down-weight the overrepresented classes so that their influence on the total loss is small while enlarging the contribution from the classes with small samples.[8] It makes the training focus on the hard examples so as to stabilize the generalization to different classes.

3.3.3 Hinge Loss. Lastly, we attempted to train the model with the hinge loss which is widely used in the semantic segmentation project. Similar to the support vector machine algorithm, hinge loss excels in binary classification by maximizing the margin between different classes. However, it led to unstable training and we did not manage to obtain meaningful results.

4 POST PROCESSING

4.1 Model performance visualisation

We will visualise specific examples to understand what kind of images the model is struggling with and we may also find if there are some bad quality data or check again the label itself if it is correct by professional pathologists. Here, we showed the prediction of random samples/ samples with the largest loss/ samples with the smallest loss. Ramprasaath proposed a technique named Grad-CAM for producing 'visual explanations' for decisions from a large class

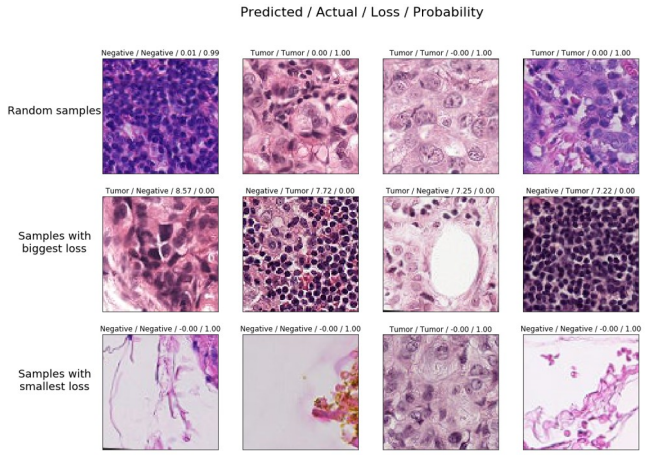

Figure 14: The prediction of random samples/ samples with largest loss/ samples with smallest loss

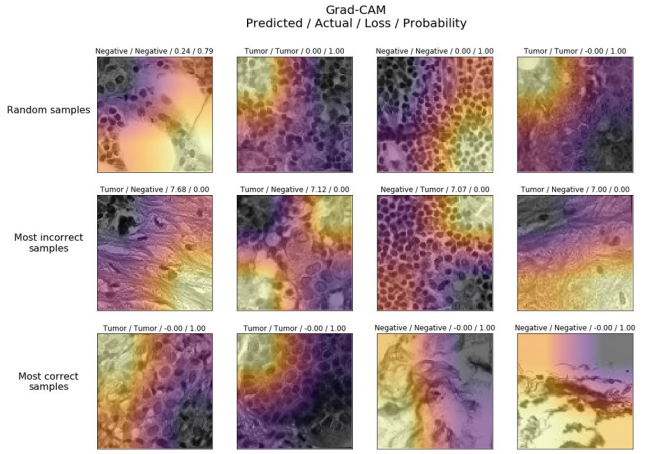

Figure 15: The heatmap of random samples/ samples with largest loss/ samples with smallest loss

of CNN-based models, making them more transparent.[9] It helps us understand by highlighting the area that the model considers important for the classification decisions. Moreover, we can also adjust our image dataset more diversely prepared, once we find the model is 'looking at' inconsequential places.

4.2 Data distillation

Unlabeled data can be used for further training to learn new feature by data augmentation, which is namely data distillation. The idea is derived from the knowledge distillation where the combined set of training dataset and the unlabeled dataset with generated labels can be used to train a student model.[10] Using the prediction of unlabeled data from the pre-trained model, the label with a high probability of accuracy can be assumed as its true label and then used to retrain the model. In order to make the model fully extract the information from pseudo-labeled data, data augmentation is

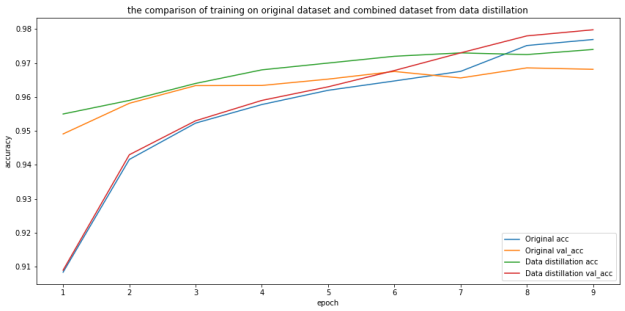

Figure 16: the comparison of training on original dataset and combined dataset from data distillation

The figure shows that data distillation boosts the performance of both training-set accuracy and validation accuracy. In the first few epochs, the difference is not obvious between them, but the gap becomes larger with the epoch rises. Finally, both training-set and validation accuracy achieve above 97.5%.

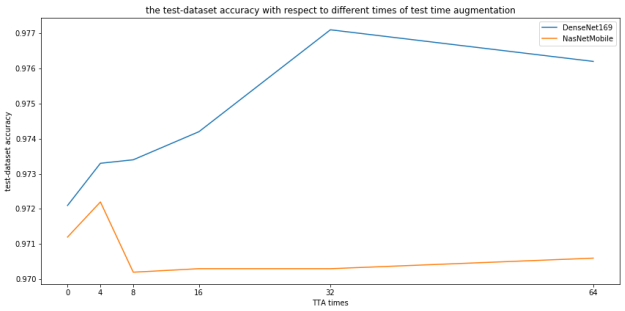

Figure 17: the test-dataset accuracy with respect to different times of test time augmentation

The figure shows that test time augmentation cannot keep the prediction accuracy improving with the time rises. The peak is different in different models. For example, the DenseNet169 model can achieve 97.77% accuracy with 32 times of TTA, which is much higher than the peak of NasNetMobile (97.2%). It represents that DenseNet169 has a better ability of feature extraction than NasNetMobile.

introduced to emphasize specific feature and generate the variants, which is of benefit to improve the robustness. The author of data distillation also found that retraining the model with pseudo-labeled data results in better performance than fine-tuning the pre-trained model. Furthermore, a combination of pseudo-labeled data and ground-truth data in each batch would improve gradient estimates in the training process. With the use of these tricks, the accuracy of our cancer classification model is boosted from 97.64% to 97.80%.

4.3 Test Time Augmentation

During test-dataset prediction, we resize the images into the 224x224 size and normalize RGB channels the same with training in advance. Similar to the data augmentation in the training process, Test Time Augmentation (TTA) is to apply several transformations to the test image and then generate a different prediction for it. Therefore, it would take full advantage of the feature in the test image instead of inputting the original image to the model directly. The predictions of corresponding modified images are averaged and then output as the final outcome. It is often superior to the prediction from the image with a single augmentation.

4.4 Snapshot ensemble

In order to accelerate the training, we often train the model on a large learning rate at the beginning stage. When it is close to a solution, the learning rate would be dropped to find the closest minimum. Although the training process for the neural network would never reach a global minimum, the local minimum that we get during the training leads to a great difference in generalization. The purpose of the snapshot ensemble is to make full use of these local minima to improve the overall generalization. Instead of retraining the model from scratch, Huang et al. (2017) argue that using Cosine annealing which repeatedly adjusts the learning rate from large to small would contribute to escaping the current local minimum and then converging into another solution.[14] The weights of each local minimum are saved and our final prediction is an ensemble consisting of different snapshots in the optimization process.

4.5 Multiple models fusion

By averaging the results of multi-model prediction, it is often possible to obtain a better prediction than any of the model predictions, indicating that they can contribute new information which is not extracted by a single model. However, the result of the combination would be worse when the models have great differences in performance.

Models fusion	train-set acc	test-set acc
DenseNet169(5fold CV)	97.66%	97.71%
DenseNet169+Xception	97.56%	97.42%
DenseNet169+NasNet mobile	97.28%	96.88%
DenseNet169(Snapshots ensemble)	97.33%	97.32%
NasNet mobile+Xception	97.23%	97.10%

REFERENCES

[1] Bergstra, J., and Bengio, Y. hyper-parameter optimization. In *Journal of Machine Learning Research* (2012), pp. 281–305.

[2] Chollet, F. Xception: Deep learning with depthwise separable convolutions. *CoRR abs/1610.02357* (2016).

[3] D A, Salazar. J I, V. Comparison between svm and logistic regression: Which one is better to discriminate? In *Revista Colombiana de EstadÃŋstica* (2012), pp. 223–237.

[4] He, T. Bag of tricks for image classification with convolutional neural networks. In *Amazon Web Services* (2018), p. 3.

[5] Huang, G., Liu, Z., and Weinberger, K. Q. Densely connected convolutional networks. *CoRR abs/1608.06993* (2016).

[6] Krizhevsky, A. Imagenet classification with deep convolutionalneural networks. p. 1.

[7] Leslie, N. S. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. p. 146.

[8] Lin, T.-Y. Focal loss for dense object detection. IEEE Computer Society Conference, p. 7.

[9] R. Selvaraju, R. Grad-cam: Visual explanations from deep networks via gradient-based localization.

[10] Radosavovic, I. Data distillation: Towards omni-supervised learning. p. 10.

[11] Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. Regularized evolution for image classifier architecture search. *CoRR abs/1802.01548* (2018).

[12] Schroff, F. Facenet: A unified embedding for face recognition and clustering. IEEE Computer Society Conference, p. 7.

[13] Szegedy, C., Ioffe, S., and Vanhoucke, V. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR abs/1602.07261* (2016).

[14] Zhuang, L. Snapshot ensembles: Train 1, get m for free. p. 5.