Recognition of Oracle Bone Characters

Jia Wei Ngo, Student ID: 717030990005

***Abstract*—This paper covers two different methods that can be used for an image classification problem, a Convolutional Neural Network (CNN) and a Bag of Visual Word (BOVW) model generated by k-means clustering with Speeded up robust features (SURF). We also investigate the benefits from Data Augmentation when a limited dataset is provided.**

***Index Terms*—Convolutional Neural Networks, Bag of Visual Words.**

I. Introduction

IMAGE classification refers to the labelling of images into one of a number of predefined categories. Humans classify images on a daily basis. From identifying that an image depicts the Eiffel tower in Paris to reading words, this is a trivial task for humans. However, it has proved to be a complex problem for computers [1].

There are three major techniques in image classification. Supervised classification, unsupervised classification and semi-supervised classification [2]. In this paper, we investigate two supervised classification methods, CNNs and a BOVW from SURF.

II. Problem Statement

A small dataset (85 images per class) of handwriten oracle bone characters are provided, belonging to a total of 40 classes as shown in Figure 1.

Fig. 1. Some of the oracle bone characters provided

We investigate different methods for this recognition task.

III. Convolutional Neural Networks

For Computer Vision problems like image classification, approaches such as SIFT/SURF were traditionally used, with particular success in representations involving BOVW descriptors [3].

The idea of CNNs dated back as early as 1998 [3], working well for hand-written digit recognition [4], which is similar to our problem. CNNs are favourable for image classification problems due to the limitations traditional neural networks will face - the matrix of input weights would be too high [4].

The model that we have adopted for this paper is shown in Figure 2.

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 32)        896
_________________________________________________________________
activation (Activation)      (None, 48, 48, 32)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 24, 24, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 24, 24, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 22, 22, 64)        18496
_________________________________________________________________
activation_1 (Activation)    (None, 22, 22, 64)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 11, 11, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 11, 11, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 9, 9, 64)          36928
_________________________________________________________________
activation_2 (Activation)    (None, 9, 9, 64)          0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 64)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 4, 4, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 1024)              0
_________________________________________________________________
dense (Dense)                (None, 128)               131200
_________________________________________________________________
activation_3 (Activation)    (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               16512
_________________________________________________________________
activation_4 (Activation)    (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 40)                5160
_________________________________________________________________
activation_5 (Activation)    (None, 40)                0
=================================================================
Total params: 209,192
Trainable params: 209,192
Non-trainable params: 0
_________________________________________________________________
```

Fig. 2. Tensorflow Summary of model used for CNN

A. 40 Class Model without Data Augmentation

A total of 3440 images were provided in the dataset, with 85 images per class as stated in II. The CNN in Figure 2 was first tested without any form of data augmentation on the dataset. After training the model, the results can be seen in Figure 3. During training of the CNN model, 20% of the dataset was used for validation, with 80% as the training set. From Figure 3, it is observed that overfitting has occurred, from the Training and Validation loss. After the 30th epoch, the Validation loss has started to rise, increasing the difference between Training Loss and Validation Loss, a sign of overfitting.

A limited dataset is one of the causes of overfitting [5]. Several methods may be used to deal with overfitting like

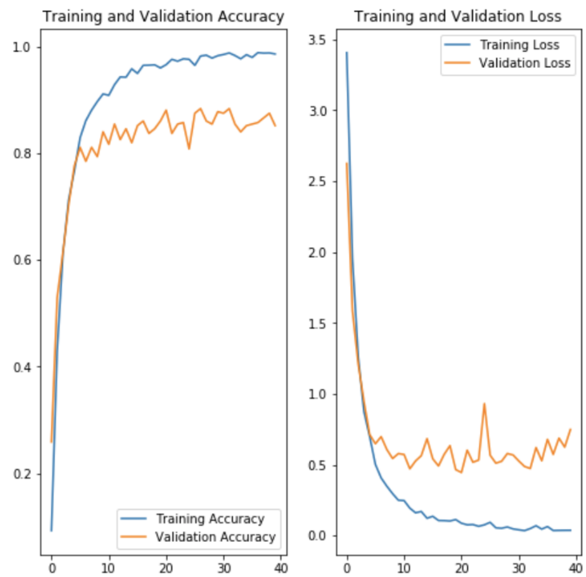

Fig. 3. Training and Validation Acc and Loss; 40 classes no data augmentation

implementing Dropout [5], Early Stopping and increasing the dataset Data Augmentation.

B. 40 Class Model with Data Augmentation

There are several benefits to data augmentation, with a small dataset. Without data augmentation, the average validation accuracy we were able to achieve was **86%**. Data augmentation may help to prevent overfitting by increasing the size of our dataset and improve the generalization of the model with more diverse data, improving accuracy [6].

Figure 5 shows the snippet of code used for Data Augmentation of the dataset. Several methods were used, including rotation, scaling, horizontal and vertical translation.

With Data Augmentation, we were able to increase the dataset of 3440 to a total of 31228, split evenly among classes. The results from training the CNN are shown in Figure 6.

It is observed that the difference between training and validation accuracy/loss has decreased, and our validation accuracy has improved to **92%** from **86%**. The benefits of data augmentation have been observed.

Figure 4 shows the confusion matrix of the model on the validation set of images.

C. 10 Class Model with Data Augmentation

Apart from the 40 Class Model, we also classify the oracle bone characters based on their meaning. For instance, in this case, 4 variant words are provided for words of a certain meaning.

The results from the 10 Class Model with similar Data Augmentation done are shown in Figure 8. Validation accuracy from the 10 Class Model is **97%**.

Figure 7 shows the confusion matrix of the 10 Class Model on the validation set of images.

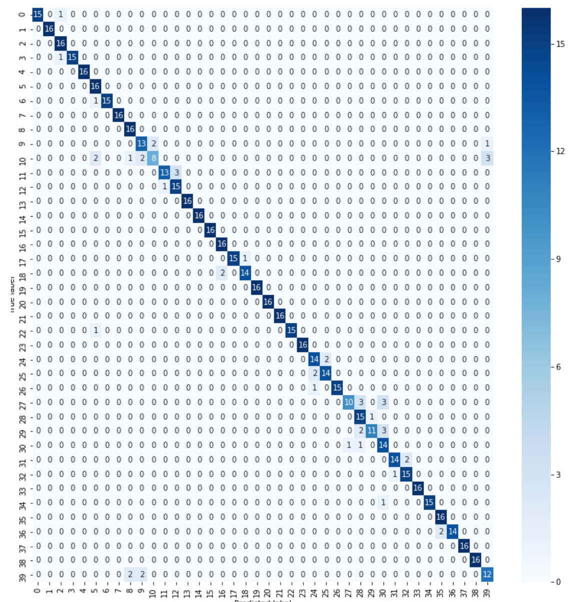

Fig. 4. Confusion Matrix of CNN; 40 classes with data augmentation

```
from keras.preprocessing.image import
    ImageDataGenerator

aug = ImageDataGenerator(
            rescale = 1./255,
            rotation_range = 20,
            width_shift_range = 0.10,
            height_shift_range = 0.10,
            zoom_range= 0.05
            )
```

Fig. 5. Snippet of code used for augmentation

Fig. 6. Training and Validation Acc and Loss; 40 classes with data augmentation

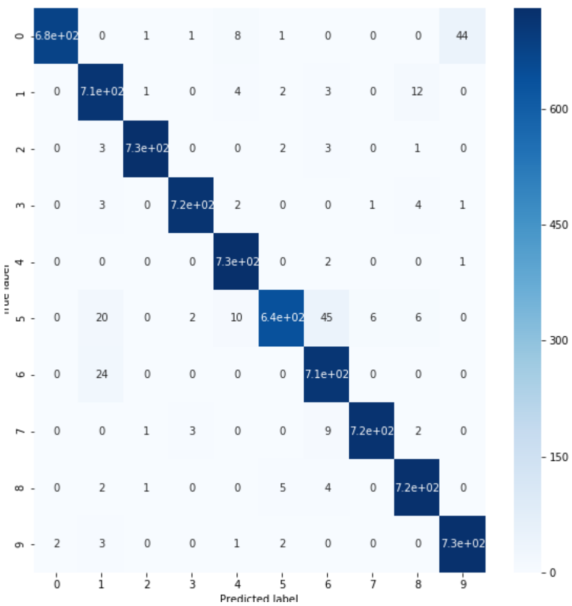

Fig. 7. Confusion Matrix of CNN; 10 classes with data augmentation

Fig. 8. training and validation acc and loss; 10 classes with data augmentation

D. Testing the models

Figure 9 shows the 8 hand-drawn characters used for testing, taken in different scenes to test the robustness of the CNN model generated. These images were not used in training the model. From the images, we can observe that the images are taken in different conditions. The bottom row images were taken in low-light scenarios, which explains the noise around the oracle bone character. A different coloured paper was also used.

Figures 10 and 11 show the prediction results of the 8 images used in Figure 9, on the 40 Class and 10 Class Model respectively.

From the results, we observe that the colour of the background did not significantly affect the prediction confidence of the models. In the 40 Class Model, there are some images which produced lower confidence levels. This is largely due to the similarities with their variant words. The high confidence levels in the 10 Class Model shows that the proposed models are able to work well regardless of different lighting, ink used as well as colour of paper used.

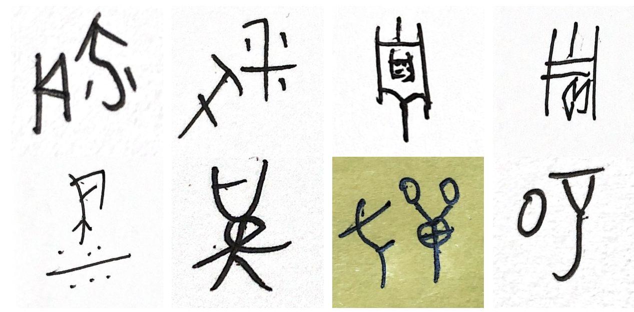

Fig. 9. 8 hand-drawn oracle bone characters, with differing lighting, pen width and colour of paper; From left to right: class 5, 6, 12, 14, 23, 28, 33, 36 respectively

```
Predicted class_5 with 0.9990446 confidence.
Predicted class_6 with 0.94465 confidence.
Predicted class_12 with 0.99898773 confidence.
Predicted class_14 with 0.9806209 confidence.
Predicted class_23 with 0.9838792 confidence.
Predicted class_28 with 0.81984067 confidence.
Predicted class_33 with 0.98514426 confidence.
Predicted class_36 with 0.7313329 confidence.
```

Fig. 10. Results of test images on 40 class model

IV. Bag of Visual Words approach

Experimental setup for the BoVW approach was done in MATLAB. In the MATLAB implementation, a bagOfFeatures object is created by extracting the SURF descriptors from images in the dataset and performing k-means clustering, returning a visual word histogram for each image in the dataset.

Figure 12 shows an example of a visual word histogram generated for one of the images.

A. 40 Class Model no Data Augmentation

The training set consists of 2400 oracle bone character images split into 40 categories and the test set consists of 1040 of such images (total of 3440 images as provided).

Running the categoryClassifier on the test set returns an average accuracy of **0.90** (Mean of the diagonal of confusion matrix).

Figure 13 shows the confusion matrix of the 40 class categoryClassifier evaluating the test set.

```
Predicted class_1 with 0.9909314 confidence.
Predicted class_1 with 0.99999285 confidence.
Predicted class_3 with 1.0 confidence.
Predicted class_3 with 1.0 confidence.
Predicted class_6 with 0.9999999 confidence.
Predicted class_7 with 1.0 confidence.
Predicted class_8 with 0.9987197 confidence.
Predicted class_9 with 0.9996848 confidence.
```

Fig. 11. Results of test images on 10 class model

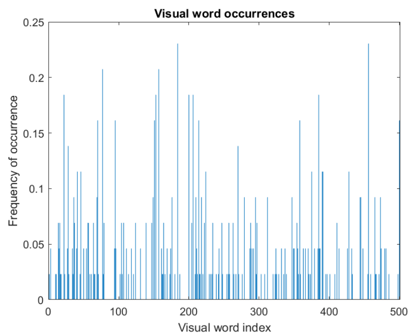

Fig. 12. Visual word occurrences histogram generated for an example image

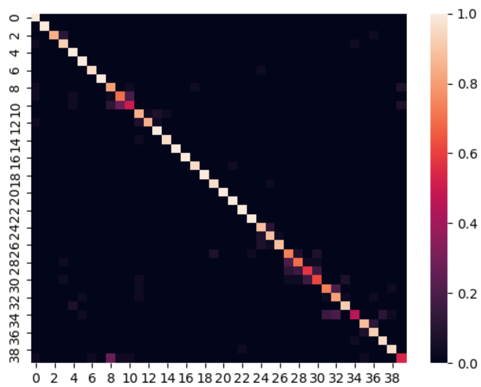

Fig. 13. Confusion Matrix of BoVW; 40 classes without data augmentation

Similar to the CNN method, the BoVW CategoryClassifier had some trouble with classes that had similar features with other classes.

B. 10 Class Model no Data Augmentation

The training set consists of 2410 oracle bone character images split into 10 categories and the test set consists of 1030 of such images (total of 3440 images as provided).

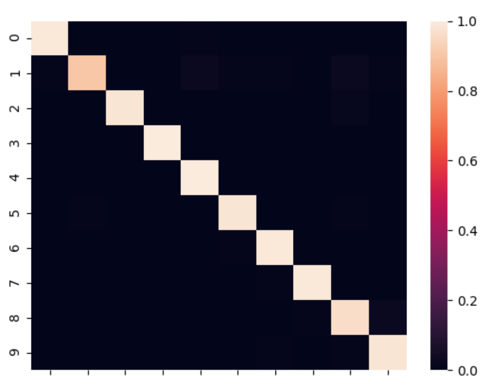

Fig. 14. Confusion Matrix of BoVW; 10 classes without data augmentation

Running the categoryClassifier on the test set returns an average accuracy of **0.98** (Mean of the diagonal of confusion matrix).

Figure 14 shows the confusion matrix of the 40 class categoryClassifier evaluating the test set.

C. Testing the models

Using the same 8 hand-drawn characters in Figure 9, the BoVW models struggled to predict the image classes correctly. In the 40 class model, only 4 out of 8 images were classified correctly, while the 10 class model classified 6 out of 8 images correctly.

From these results, we may observe that the CNN is more generalised towards new data that are not necessarily of the same conditions.

Although the confusion matrix on unseen test data did well, the model was not able to generalise as well as the CNN to new unseen data of different conditions.

V. Conclusion

In our experiments, the CNN method provided better generalisability than the BoVW method. This could be due to the nature of the classification and dataset used. Although, BoVW methods have seen great success in Content Based Information Retrieval (CBIR) systems [7].

Some further experiments could be done to potentially improve the results of the BoVW method, such as testing different feature extraction methods such as Scale-invariant feature transform (SIFT) or Oriented fast and Rotated brief (ORB).

The benefits of data augmentation have also clearly been demonstrated when a limited dataset is provided.

References

[1] P. Kamavisdar, S. Saluja, and S. Agrawal, "A survey on image classification approaches and techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 1, pp. 1005–1009, 2013.

[2] M. Jain and P. S. Tomar, "Review of image classification methods and techniques," 2013.

[3] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. Kruthiventi, and R. V. Babu, "Chapter 2 - an introduction to deep convolutional neural nets for computer vision," in *Deep Learning for Medical Image Analysis*, S. K. Zhou, H. Greenspan, and D. Shen, Eds. Academic Press, 2017, pp. 25 – 52. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128104088000031

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[6] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *CoRR*, vol. abs/1712.04621, 2017. [Online]. Available: http://arxiv.org/abs/1712.04621

[7] A. Alfanindya, N. Hashim, and C. Eswaran, "Content based image retrieval and classification using speeded-up robust features (surf) and grouped bag-of-visual-words (gbovw)," in *2013 International Conference on Technology, Informatics, Management, Engineering and Environment*. IEEE, 2013, pp. 77–82.