Removing Noise from Hand-Written Transcriptions

University of St Andrews

Junichi Hattori

17, Aug, 2021

Abstract

The historical documents written by ancient Egyptian hieroglyphs utilising OCR tool has shading to indicate damaged text. This project is to detect diagonal lines and remove them and we use the approach which is the combination of discrete Fourier Transform(DFT)/fast Fourier Transform(FFT) and Support Vector Machine(SVM) to deal with this issue. There are two conditions of grid size for applying DFT/FFT to achieve the high performance. First is the grid size is small and second is to enhance the resolution such as merging pixels. As a result, the numbers of misclassified labels are less and the grid of 6x6 reaches about 95% accuracy. Thus, the combination of DFT/FFT and the supervised learning has a contribution to detect diagonal lines.

Declaration

"We declare that the material submitted for assessment is our own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated.
"The main text of this project report is 6480 words long, including project specification and plan.
"In submitting this project report to the University of St Andrews, We give permission for it to be made available for use in accordance with the regulations of the University Library. We also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. We retain the copyright in this work."

Acknowledgement

I would like to thank my supervisor Dr Mark-Jan Nederhof for his most valuable guidance, insightful comments and boundless encouragements during this research project. The master's project would not have come to completion without his kind support. Thank you very much for not giving up on me through the slower than expected process and for being a role model both academically and personally.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

This section is about then introduction of this project. It consists of the study of ancient Egyptian hieroglyphs, the purpose of dissertation. In the end, the project structures are mentioned.

1.1 Background

Ancient Egyptian hieroglyphs is the key role to learn the culture of ancient Egyptians. Thus, the boards hieroglyphs are inscribed on have kept and preserved over the centuries. However, these boards are fragile, and it is necessary to encode the hieroglyphs. The Optical Character Recognition (OCR) is the designed tool to recognise the characters in the images and generate the normalised encoding. In this project, we use the encoded historical documents utilising OCR. This process is what scan the documents and converting machine readable format. However, a diagonal line, which is so called shading or hatching includes in the historical documents to indicate damaged area. Although the researchers can attempt to remove diagonal lines manually, it is monotonous and tedious long work because there are plenty of shadings in the historical documents. In addition to that, expert knowledge is required to judge whether part of shading or part of symbol in order to avoid human errors. Therefore, to fix this issue, this project focuses on developing approaches. The approach in this project is the supervised learning.

1.2 Aims and Objectives

The main goal of the project is to remove hatching from handwritten characters. However, it is necessary to break down into more specific objectives. The objectives of the project are as follows:

1.2.1 Primary Objectives

- Identify the hatching superimposing on the images of ancient Egyptian hieroglyphs utilising FFT algorithm
- Create the dataset for machine learning and develop an tool to generate dataset automatically
- Remove the hatching from the images of ancient Egyptian hieroglyphs based on the result of analysing FFT values
- Repair the parts of symbols which are classified as shading incorrectly

1.2.2 Secondary Objectives

- Evaluate the performance of other dataset aside from images in the project
- Tune the hyperparameters to optimise the network structures
- Attempt alternative techniques in terms of this issues and compare the performance with primary suggested one

1.3 Achievements

The extent of my achievement in this project is described in this section. We indicate to apply the combination of discrete Fourier Transform(DFT)/fast Fourier Transform(FFT) and Support Vector Machine(SVM), which is a type of the supervised learning method into test set to detect diagonal lines. In the case that the size of grid for DFT/FFT is 6x6 and merging 4pixels into one to enhance resolution, most diagonal lines are detected and removed. The accuracy is 94.49% and it is effective to add DFT/FFT values into features. However, the images in the training and test set are created artificially. Thus, we consider further research to apply this approach in practice.

1.4 Project Structures

This report consists of six chapters. The first chapter is the current one. Chapter 2 introduces the literature reviews of this project. It describes the importance of ancient Egyptians' hieroglyphs and one of the texts is used as dataset in this research. Then, as to noise, which is a key factor of this project, it presents from typical noises to existing approaches for detecting and removing them. Some approaches are employed to this project in order to be effective in terms of removing diagonal lines. In the chapter 3, it proposes the research methodology of removing diagonal lines. It breaks down in to three parts, the design of creating dataset, the approach of detecting hatching texts and removing them. The chapter 4 is about the procedures of implementing approaches based on the chapter 3. The chapter 5 summarises the performance utilising the approach mentioned in the chapter 3, 4 and indicates the results of images. This approach proves more effective to removing hatching from handwritten texts. Thus, this approach has room for improvement along the results are addressed. In addition to that, compared to the evaluation of a model applying neural network, the discussion of the results are mentioned. Finally, the chapter 6 is about the conclusion of this project and further research in the future. The reference list is all papers with the citations. Appendices presents the information on using guidance and ethics self-assessment form.

Chapter 2

Literature Review

In this chapter, we explain about the reason why we reach the purpose of the project. This chapter will introduce Ancient Egyptian hieroglyphs, and also some kinds of noise that can occur in images. After that, we review the related work regarding removing noise using both fast the Fourier Transform (FFT) algorithm and machine learning. The literature review is divided into four parts - Ancient Egyptian hieroglyphs, type of noise, the fast Fourier Transform algorithm and machine learning.

2.1 Ancient Egyptian Hieroglyphs

The Egyptian civilisation is one of the oldest and most mysterious. civilization is mysterious. The ancient characters in Egypt are unique and consist of combinations of logographic, syllabic and alphabetic elements. Thus, they are hard for modern people to comprehend. Deciphering arcane characters is the only way to understand this ancient culture and it contributes to the development of archaeology. Thanks to the great efforts of archaeologists, we can now read these ancient texts. However, Egyptian is already a dead language and there are still undeciphered texts. Therefore, our knowledge of this ancient civilization is limited, and we should preserve ancient relics. The boards the ancient characters are inscribed on is made of stone and they are fragile. Sethe (1927) [1] transcribed one of the ancient texts by hand to protect these fragile stones. In the texts of handwriting by Sethe, he added hatching to indicate damaged text regarding some specific parts (Nederhof, 2015) [8] and these hatchings are obstacles to deciphering the details of ancient culture and removing this 'noise' is challenge. We can remove hatching manually, using existing tools but it is time-consuming process. In this research we will consider methods for removing this noise automatically.

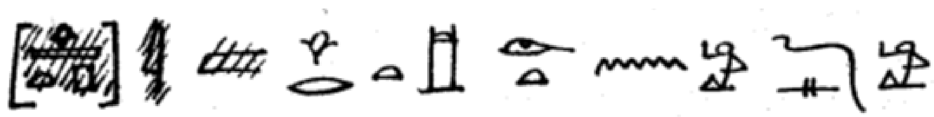


Fig 2.1.1: Damaged text with hatching [1]

2.2 Type of Noise

Noise is random signal and there are various kinds of noise in image processing. In this section, let me give you some typical noises. Gaussian noise is the electrical noise which arises in amplifiers and detectors Natural source such as thermal vibration of atoms and discrete nature of radiation of warm objects brings to the reason of causing it. Also, Gaussian noise is obstacles the gray value in digital image and that is characteristics by PDF or normalises histogram(Boyat and Joshi, 2015) [3].
The impulse valued noise is the noise that black and white dots includes in the image, and is called salt and pepper noise. This noise is that the original data in some parts of pixels drop because of sharp and sudden change in image signals (Boyat and Joshi, 2015, Patro and Panda, 2016) [3, 9].

(a)

254	207	210
97	212	32
62	106	20

254	207	210
97	0	32
62	106	20

(b)

Fig 2.2.1: (a)Image with Gaussian noise [2], (b)The central pixel value is corrupted by Pepper noise [3]

In this project, the hatching (also referred as shading) to indicate damaged text is called as noise and we attempt to remove it. The hatching which overfit images has diagonal lines, but line spacings are not uniform and gradients of lines also different respectively because of handwriting.

2.3 Fast Fourier Transform

Image recognition is necessary before denoising and a lot of research has been carried out to do that regarding handwritten image. Kaur and Singh(2016) [4] evaluated the accuracy of Gurmukhi handwritten characters. They used two type of features, spatiotemporal and frequency(FFT) to train three kinds of classifiers, K-Nearest Neighbour(KNN), Multilayer Perception(MLP) and Support Vector Machine(SVM). Also, both tenfold cross validation and percentage split methods are used. When focusing on SVM with spectral features, the accuracy of recognition is less than other method and spatiotemporal feature, but that is about 85% and we can use this method to a certain extent as the way of recognising handwritten characters.

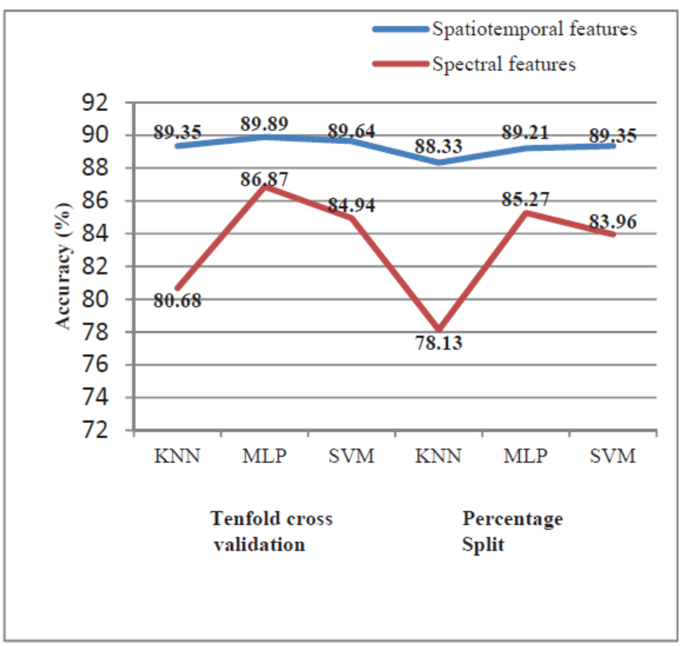
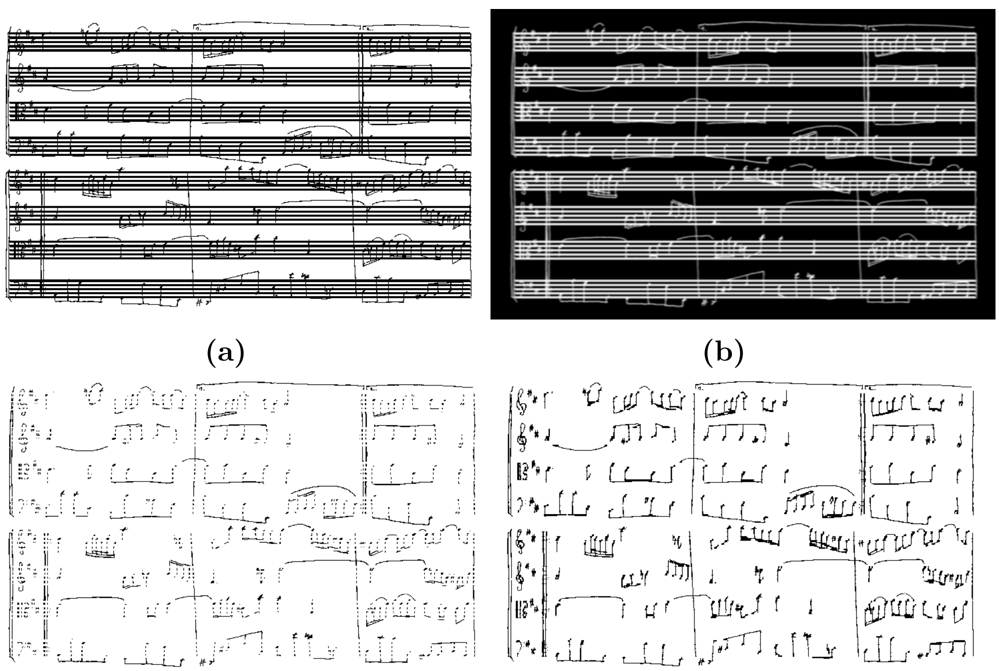


Fig 2.3.1: Comparison of accuracy for tenfold and percentage split methods [4]

Also, the researchers who attempt to remove handwritten noise using FFT and Gaussian Lowpass filter is Alirezazadeh and Ahmadzadeh(2014) [5]. The approaches of many researchers are limited for images such as only horizontal and vertical staff lines, but their approach can detect handwritten staffs and remove them. That means that it is effective and robust under deformations even if the staff lines are not straight lines. The shadings in the hieroglyphs are handwritten diagonal lines and all of them are not the same slopes. Thus, this approach using FFT and Lowpass supports detecting the shading text.



(a) **(b)**

(c) **(d)**

Fig 2.3.2: The Procedure of Staff Removal. (a) The Input Image. (b) Applying FFT and Lowpass Filter. (c) The Image Obtained after Removing Staff Lines Completely. (d) The Output Image. [5]

2.4 Machine Learning

As for various noise which are introduced in the section 2.2, there are some approaches to remove them.
Ghose et.al(2020) [2] indicates that applying CNN model regarding image denoising is the best performance compared to other methods. They create dataset adding the Gaussian noise to the image without including noise. They evaluate various methods, Wiener filtering, Bilateral filtering, Principal Component Analysis (PCA), Wavelet based transform method and CNN based denoising. As a result of evaluating three metrics(table1), CNN based denoising is the better than all other methods and it shows the effective of CNN based denoising.

Image Denoising Methods	**Image quality assessment Parameters**		
	MSE	***PSNR***	***SSIM***
Wiener filtering	84.9830	25.0098	0.7872
Bilateral filtering	83.6294	25.9021	0.7982
PCA	80.9892	26.7834	0.8345
Wavelet based transform method	81.8765	26.1243	0.8181
Total Variation based regularization	81.3920	27.9898	0.9087
CNN based denoising	**78.4622**	**28.7126**	**0.9106**

Table 2.4.1: Quantitative Analysis [2]

As the similar approach of Ghose et.al(2020) [2],Komatsu and Gobsalves(2018) [6] proposes U-Net which is a kind of Convolution Neural Network(CNN) to remove noise from handwritten character images. In terms of processing input, they also the same as well as Ghose et.al(2020) [2]. They add various kinds of noise such as Gaussian noise, blur and mix to original images of a handwritten character and these images are as input. The loss between input and output is few when the number of epoch of training cycle is set to 100 and they can succeed in removing noise from a hand-written character. Thus, the approach regarding removing artificial images combining noise to original images is effective. However, test data in this paper focuses on only a handwritten character not text. In addition to that, the images using test data are created artificially and we cannot judge whether this method is effective in case test images are raw data including noise.

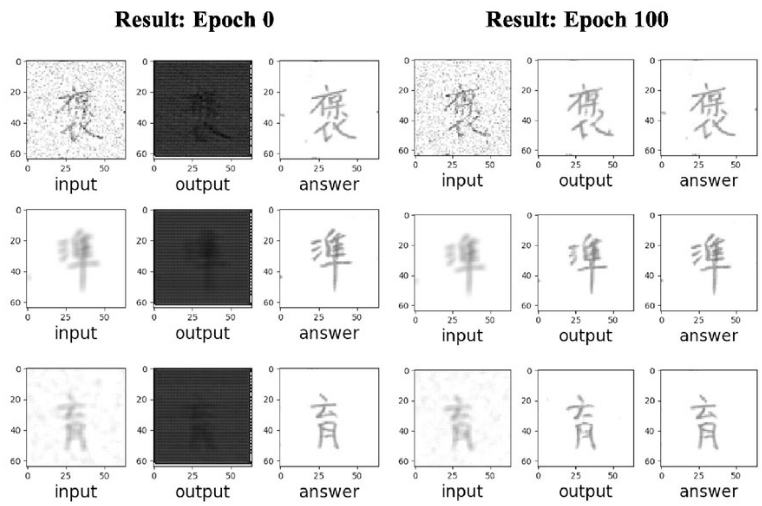
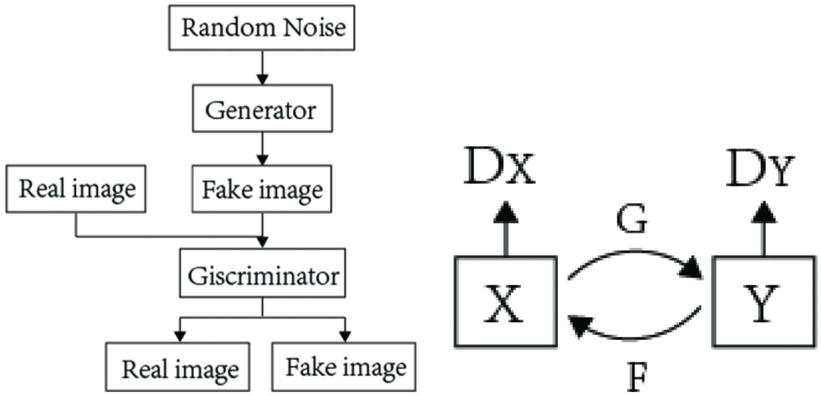


Fig 2.4.1: Test result using U-Net [6]

Although Komatsu and Gobsalves(2018) [6]'s research is supervised learning, the research of Sun and Zhang(2020) [7] utilised the method of Cycle-GAN, unsupervised learning to remove the grid background of handwritten Chinese compositions automatically. CycleGAN is the method proposed by the research team of Berkeley University on ICCV in 2017 and it is based on GAN which Goodfellow proposed. This method brings to the better result in terms of image generation, and they apply this method to remove grid background of handwritten characters. While the generators to remove the grid background utilised both ResNet and UNet, UNet is more effective than ResNet. Therefore, the application of UNet as a generator have a good effect to remove noise with handwritten with regard to both case of supervised and unsupervised learning. UNet is one of the effective methods when removing hatching.



(a) **(b)**

Fig 2.4.2: (a)Net structure of GAN [7], (b)Net structure of CycleGAN [7]

2.5 Summary

In this project, removing diagonal lines from handwritten transcriptions is the purpose and we introduced background relating to this project and approaches to remove noise. As the ways of detecting noise, FFT and SVM are effective, and they might bring to the support of detection of hatching. In addition to that, applying CNN or UNet when removing noise have a good effect as results of Ghose et.al(2020) [2],Komatsu and Gobsalves(2018) [6] and Sun and Zhang(2020) [7]. Thus, these methods might be one of the most effective ones to remove handwritten diagonal lines.

Chapter 3

Design

3.1 Introduction

In this chapter, it describes an approach and techniques used by this project. In the research, there is the approach to detect diagonal lines from the hieroglyphic texts. That is the combination of fast Fourier Transform and supervised learning classification. It involves Support Vector Machine(SVM). As to dataset, there is no existing one for detecting and removing diagonal lines. Therefore, artificial combined images are created automatically after preparing unshaded texts and shading pieces manually and divide into training data and test data. After processing that, all pixels of combined images are classified as either white, part of shading and part of text. The introduction of the values adapting discrete Fourier Transform, the amplitude and phase, as predictors and the SVM model are trained to embark on the task. With respect to the measure of evaluation, mainly F-measure is adopted. After training the model, it predicts the labels which belong to part of shading. It repairs the pixel based on the performance of detecting shading. The design of data is described first. The second section is about the way of labelling each pixel. Then, the values of fast Fourier Transform are employed to the input data. After that, it proposes SVM, the supervised learning. At last, repair process along with the result of test data are attempted and it concludes for this chapter.

3.2 Design of Preparation Training Data

In this section, it presents the introduction of the raw dataset of ancient Egyptian hieroglyphs and the method of creating the training and test dataset from these sources. In this research, the *Urkunden I* is adopted as data sources which are acquired online. These texts are handwritten-transcriptions written by Kurt Sethe. The required datasets to detect shading are the ones with shading. However, the existing images are less information regarding shaded text when learning a model. Thus, some types of unshaded texts and shading pieces are arranged and merged as shading texts. In terms of shading pieces, the length and thickness of them are different per shading pieces in order to make the model learn some kinds of shading style. After completing artificial data, we acquire the training and the test dataset. The library used in creating data is the Open Source Computer Vision Library (OpenCV). As an example of combined text is shown in the Figure3.2

3.3 Design of Input Data and Label

The supervised learning algorithm is used to solve the problem based on the predictors with training sets. Thus, classifying labels per pixel is necessary to detect the diagonal lines. This section introduces the choice of predictors.

3.3.1 Design of Input Data

The model in this research is the supervised learning and it is trained some features and its labels. The required features per pixel to learn are 'location(x,y)', 'gray value', 'discrete Fourier Transform values(amplitude, phase)' and 'label'. gray value of each pixel to distinguish white pixels from black ones. As mentioned above, binarisation process is implemented when creating combined images. Thus, gray value of each pixel is only black (gray value = 0) or white (gray value = 255). In terms of 'FFT values' and 'label, it proposes in the next subsection.

3.3.2 Fourier Transform

The location and gray value of each pixel are not enough to predict label. It is because that both part of shading and text are the same black pixels and there is no difference between them. We assume that the peripheral parts around the pixel at any coordinates includes information to specify labels. Thus, with respect to the whole periphery containing pixels at specified coordinates, the feature vector is obtained with fast Fourier Transform. According to these values, the amplitude and phase at specified coordinates are regarded as features. The formula of amplitude and phase is as below.

$$
\begin{aligned}
X(k) &= \mathcal{F}[x(n)] = \sum_{n=0}^{N-1} x(n)\exp(\frac{-j2\pi kn}{N})(k = 0, 1, \cdots, N-1) \\
&= A(k)\exp(j\Theta(n))
\end{aligned}
$$

$$
\begin{aligned}
Amplitude : A(k) &= \sqrt{Real[X(k)]^2 + Imag[X(k)]^2} \\
Phase : \Theta(n) &= \tan^{-1}\frac{Imag[X(k)]}{Real[X(k)]}
\end{aligned}
$$

The grid patterns in this research are 5X5, 7X7, 9X9, 21X21 and 31X31. We suppose that adopting larger grids such as 21X21 and 31X31 increase the accuracy to classify labels because of containing enough information rather than smaller ones. Figure 3.3.1 shows as an example of the grid pattern,5X5.

3.3.3 Design of label

Each pixel given in the position (x,y) in the images includes either white or black pixels. With regard to black pixels, it is divided into two categories, part of shading and part of symbol. Thus, the pixels of these three types, white, part of shading and part of symbol, are set as 0, 1, 2. Finally, these labels are created to all pixels in the images.

3.4 Design of Support Vector Machine

3.4.1 Supervised Learning

Machine learning (ML) is a key part of Artificial Intelligence (AI) and it is effective to predict the label for classifying categories by the model with

learning the existing data. The framework of machine learning processes the following procedures. First, the program learned the problem through the training data. Then, the evaluation is generated utilising test data. Machine learning technique has two major different types. One of them is supervised learning and the other is unsupervised learning. In this research, the method of supervised learning technique leads to the solution of detecting shading.
There are two phases in the supervised learning. The training part is the first phase. The features in the input data support to build the model such as Support Vector Machine (SVM) and Convolution Neural Network. The performance of these model depends on the volume of data and the features. The input data and its label fed into the model. The second phase is the prediction part. The purpose of this technique is to classify the new data. Compared to the features acquired the training part, features on the unobserved data are extracted. Finally, the model is adapted to predict the label to the new data. Thus, supervised learning algorithm is possible to deal with the task of detecting the diagonal lines.

3.4.2 Support Vector Machine

As mentioned in the literature review, the combination fast Fourier Transform of Support Vector Machine (SVM) is high accuracy of image recognition. Then, SVM is a powerful classification algorithm, and this project adopts this technique. Regarding parameter tuning, default parameters scikit-learn set initially are used.

3.5 Evaluation Metrics

Measuring the performance is essential to judge the appropriate model for detecting label. This section describes the precision, recall and f1-score. In case of reviewing the images visually, the degree of detecting shading is regarded as the performance of metric compared to the test images. However, visual check is not be able to evaluate quantitatively. Therefore, three standard and concise metrics are employed to deal with the issue. The first one is the accuracy of the positive predictions named precision. The formula is as below.

$$Precision = \frac{TP}{TP + FP}$$

TP and FP stand for the true positives and negative positives respectively. Focusing on shading labels, true positives are the numbers that shading pixels are correctly classified as shading. Then, false positives are the numbers that white or symbol pixels are classified incorrectly as shading. However, it only summarises the actual results.
The second one, recall is the ratio of the observations our model predicted correctly over the total predicted observations. This measure is always used with precision. FN in the following formula is false negatives. That means shading pixels are misclassified as symbol or white pixels.

$$Recall = \frac{TP}{TP + FN}$$

F1-score is to compare the performance of test sets based on the balance between precision and recall.

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 * TP}{2 * TP + FP + FN}$$

At last, the performance of better model for detecting shading is evaluated by F1-score. Thus, the higher F1-score is the better one.

3.6 Design of repair pixel

In this section, it describes the way of removing shading based on the predicted labels. After converting the labels classified as part of shading into white pixels, it restores the images.

3.7 Conclusion

In this chapter, the methods of data preparation and classified pixels are described. One of the tasks to detect shading is to determine label per pixel. Thus, each pixel in the image is classified as either white, part of shading and part of symbol. The predicted pixels as shading are converted into white ones. In order to generate features containing the information of pixels adjacent to specified pixel, the discrete Fourier Transform/fast Fourier Transform algorithm is employed. The technique to solve the task is Support

Vector Machine, a type of supervised learning. The parameters in the SVM is default to consider the best features. The evaluation of performance is based on the F1 score.

Chapter 4

Implementation

4.1 Introduction

In this chapter, it proposes the implementation of the approach designed in the chapter 3 in detail. The first subsection is generating the training and test data from the data source, *Urkunden I*. Then, discrete Fourier Transform is introduced to add features. After that, the overview of implementation of SVM model is proposed. Finally, it describes the results of evaluation metrics and visualising detecting predicted shading.

4.2 Data Source

As to data sources with shading, many types of them exist. In this project, we adopt the *Urkunden I* as a data source. It is because that the copyright of this source has already expired

4.3 Implementation of Preparation of Training Data

The first step of achieving task is to prepare training data. The procedure of creating data as follows:

1. Convert format from PDF to PNG The format scanned images is PDF (Portable Document Format) format and it converts PNG in the function convert_from_path. In order to maintain the same resolution when converting

format, the value of resolution is 300 dpi based on the size of the image and the amount of pixel.

Name	Type	Description
pdf_path	Input array	pdf file as input
dpi = num	int	image resolution

Table 4.3.1: Parameters of convert_from_path

2. Binarisation In order to distinct either white or black pixels regarding gray value specifically, binarisation is introduced to scanned images. This step is implemented by the function cv2.imread(). The parameters of this function are in the Table 4.3.2. The type is used gray scale. Then the function of cv2.threshold is indicated in the Table 4.3.3. Typically, setting the threshold value we chose arbitrarily is necessary. However, Otsu method decides the threshold value automatically based on the distribution of pixel values. Thus, the flag of threshold is both THRESH_BINARY and THRESH_OTSU to apply Otsu method. The filtering value is set to 0 because of calculating automatically using Otsu method. In this case, all images are assumed to have bimodal images.

Name	Type	Description
src	Input array	Input image
type	int	Select flag, 0: color, 1:grayscale, -1:unchanged

Table 4.3.2: Parameters of cv2.imread

Name	Type	Description
src	Input array	Input image
thresh	int	Threshold value
maxval	int	Maximum value of filtering
type	int	Select the way of thresholding techniques e.g.THRESH_BINARY and THRESH_TRUNC and etc.

Table 4.3.3: Parameters of cv2.threshold

3. Extract symbol blocks and shading pieces Owing to crop the shading pieces and symbol blocks, the existing image processing software 'irfan view'

is used. 10 shading pieces and 20 symbol blocks are extracted from the images. In terms of dimension, both of them are the same height, 140 pixels while the width of them are different.

4. Superimpose symbol blocks into shading blocks Each shading pieces are duplicated, and it is necessary to join the copied shading pieces together to fit the same size of each block. In addition to that, the following three types of files are created to learn model in the table 4.3.4. The combinations of shading blocks and symbol blocks are 200 images. Training and test data are divided by the ratio of 8:2 utilising these 200 images randomly.

File Name	Description
data*text.png	data of the symbol block
data*shading.png	data of the shading block
data*combined.png	data of the symbol block with shading

Table 4.3.4: File names of input images, *:1-200

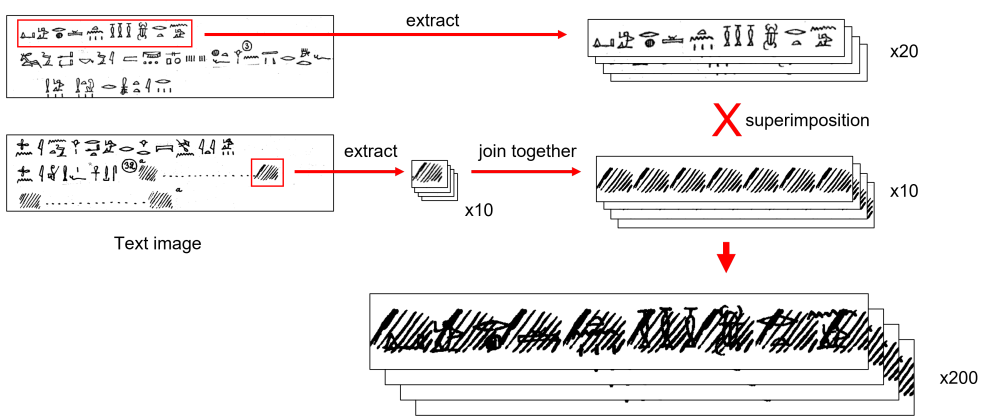


Fig 4.3.1: The way of creating combined texts

4.4 Implementation of Label

The input data is labelled following psuedo code after creating combined data.

```
For each pixel in the position (y, x) of the image with shading:
    if the value of pixel == 255 : // the value is white
      label = 0 // it is a white pixel
    else: // the value is black
       if the value of pixel == 0 and its value is in the same position (y, x) of a part of shading
       and its value is not in the same position(y, x) of a part of symbol without shading :
          label = 1 // it is a part of shading
       else :
          label = 2
         // it is a part of symbol in the case of black pixels that are both part of symbol and part of shading
```

Fig 4.4.1: classifying label

In terms of the position which belongs both a part of shading and a part of symbol, it assumes to set a part of shading.

4.5 Fourier Transform

The method of discrete Fourier Transform (DFT)/fast Fourier Transform (FFT) provides each pixel the information of assigned arbitrary grid. In order to calculate the edge pixels using DFT/FFT, we add the frame into the outside of the image in the Fig 4.5.1. This added frame changes depending on the number of grid applying DFT/FFT. Also, the values of all pixels in the added frame are 255(white). Grid patterns are $(2n + 1) \times (2n + 1)$ and $2n \times 2n$ $(n \geq 1)$. If the area applying DFT is $(2n+1) \times (2n+1)$, the added pixels to the top and the bottom are the same (n1 = n2).In case that the grid is $2n \times 2n$, the inserted pixels to the both are different(e.g. n1 = 2, n2 = 3 in case of grid 6×6). The patterns implemented in this project are as

Name	Type	Description
image	Input array	Input image
N	int	2n/2n+1 is the size of DFT/FFT grid

Table 4.5.1: Parameters of addframe/addframe update

below.

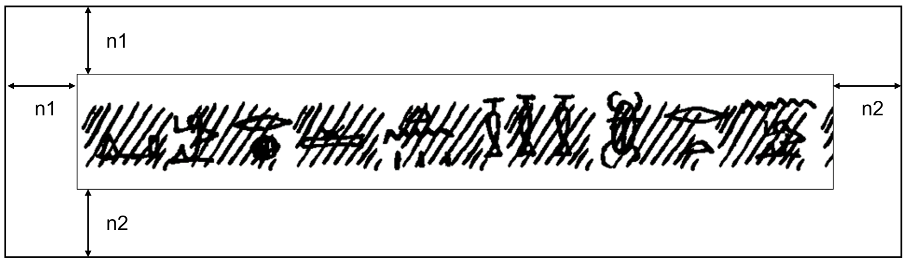


Fig 4.5.1: add frame

4.5.1 the number of grid

the number is odd

5×5, 21×21, 31×31 The result of centre position of grid applying FFT is extracted as a feature.

the number is even

The grid pattern is 6×6, 32×32. In this case, we choose 4pixels of centre of the pictures and merge into one as a feature.

4.5.2 the distance to the edge

Regarding to the symbols and shadings, we separate two labels which are either inside or edge to check whether this approach is effective. In this case, the grid patterns are 5×5 and 31×31.

4.6 Support Vector Machine

The model is constructed utilising Support Vector Machine (SVM) after preparing input data. The parameters of SVM are default ones in the table 4.6.1.

Name	Type	Default
C	float	1.0
kernel	linear, poly, rbf, sigmoid, precomputed	rbf
gamma	float	scale

Table 4.6.1: HyperParameters of svm.SVC

4.7 Implementation of Performance Metric

F_1 score is the measurement of performance on test set, which is unseen data. It is implemented by the function classification_report() which compared atual test label with predicted test label and returns the score.

4.8 Implementation of Removal task

Based on the predicted label of test image, we implement to remove hatching. After specifying the pixels which is predicted as shading, it converts these into white colour. Then, the pixels regarded as part of symbol are assigned as black colour. After determining the colour, we recover image using cv2.imwrite(). The format type is 'png'.

4.9 Run from Terminal

This section introduces the procedure of implementation in this project. First, type python3(jupyter notebook) CreateTrainingDataset.ipynb to convert 'pdf' format into 'png'format. Second, we prepare symbol blocks and shading pieces utilising the existing tool. Then, implement the same program as first procedure to create combined blocks. Next, type python3(jupyter notebook) FFT.ipynb to add the feature applying DFT/FFT into input data and label each pixel. After that, type python3(jupyter notebook) SVM.ipynb to build SVM model using training dataset and evaluate test dataset by its model. The return of this function is classification report (F_1score, accuracy and recall per test image), visualised images for classifying either each pixel is correct and images removing hatching based on predicted label.

Chapter 5

Evaluation and critical appraisal

5.1 Introduction

Evaluation of the machine learning model is a key role to judge how the model classifies the unknown data. In addition to that, it is useful when comparing other models adjusting different parameters. In this chapter, DFT/FFT is effective to remove hatching evaluating different grid patterns applying DFT/FFT.

5.2 Result of SVM

According to the grid pattern mentioned as the design and implementation respectively, the output image is as follows. Although test datasets are 40 in this project, we focus on a specify sample, 'data68combined' and evaluate it. The results based on the grid pattern indicated in the chapter.4 are from Fig 5.2.1 to Fig5.2.5.The top is a raw image(data68combined). The second image is visualised image colouring each pixel having predicted label. White is correct background colour. Black is classified as part of symbol correctly. Red is labelled as part of shading accurately. Cyan is misclassified as part of shading while the actual label is part of shading. The colour of orange is labelled as part of symbol by mistake despite the black pixel is part of shading. The third image is the output removing black pixels whose predicted labels are part of shadings based on the second image. The last image is the result removing black pixels classified as part of shading in the second image.

5.2.1 Results of the number of grid

the number is odd

The result images are Fig 5.2.1, Fig 5.2.2 and Fig 5.2.3. Total accuracy and the evaluation per label are in the table 5.2.1 - 5.2.4.

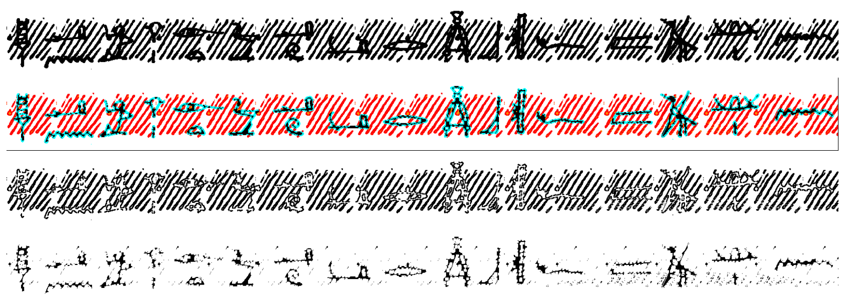


Fig 5.2.1: An example of 5x5 grid applying DFT
(top: raw,
second: the image visualised labels,
third: removing the black pixels predicted as part of symbol,
bottom: removing the black pixels predicted as part of shading)

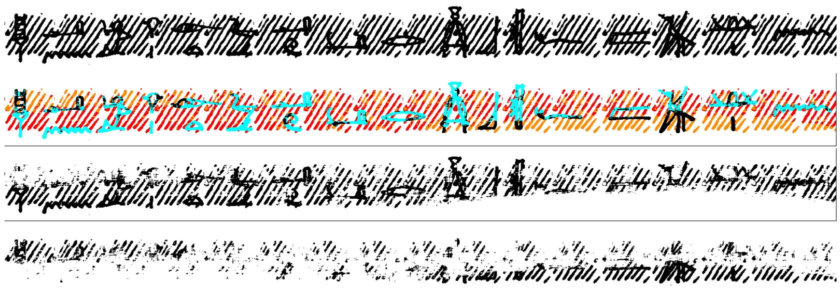


Fig 5.2.2: An example of 21x21 grid applying DFT
(top: raw,
second: the image visualised labels,
third: removing the black pixels predicted as part of symbol,
bottom: removing the black pixels predicted as part of shading)

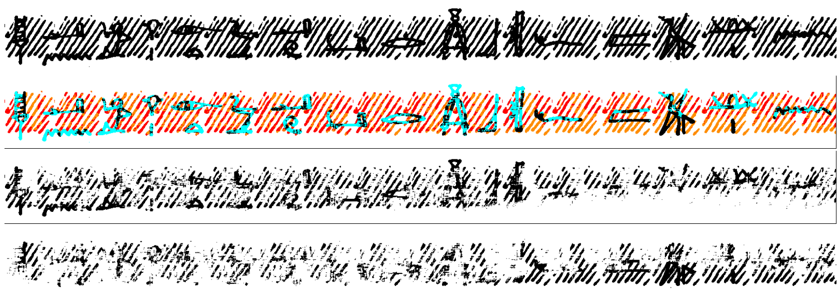


Fig 5.2.3: An example of 31x31 grid applying DFT
(top: raw,
second: the image visualised labels,
third: removing the black pixels predicted as part of symbol,
bottom: removing the black pixels predicted as part of shading)

Grid	Precision	Recall	F1-Score
5x5	0.9387	0.9387	0.9387
21x21	0.861	0.861	0.861
31x31	0.8562	0.8562	0.8562

Table 5.2.1: The accuracy of model with grid patterns applying DFT on test set

Grid	Precision	Recall	F1-Score
5x5	1	1	1
21x21	1	1	1
31x31	1	1	1

Table 5.2.2: The results model with grid patterns applying DFT regarding white label on test set

Grid	Precision	Recall	F1-Score
5x5	0.7653	0.9484	0.8471
21x21	0.6131	0.6062	0.6096
31x31	0.6167	0.5204	0.5645

Table 5.2.3: The results of model with grid patterns applying DFT regarding shading label on test set

Grid	Precision	Recall	F1-Score
5x5	0.8334	0.47	0.601
21x21	0.2968	0.3029	0.2999
31x31	0.3196	0.4106	0.3595

Table 5.2.4: The results of model with grid patterns applying DFT regarding symbol label on test set

the number is even

The result images are Fig 5.2.4, Fig 5.2.5. Total accuracy and the evaluation per label are in the table 5.2.5 - 5.2.8.

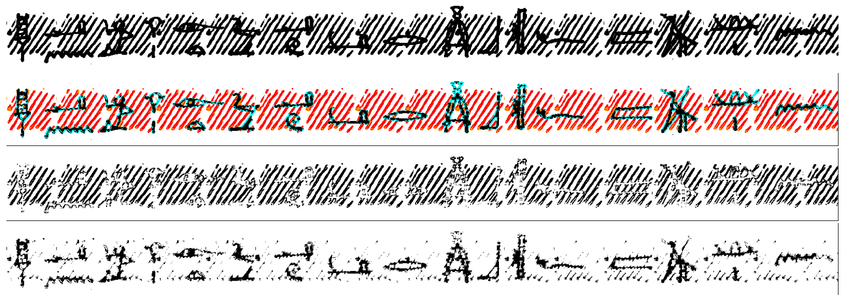


Fig 5.2.4: An example of 6x6 grid applying FFT
(top: raw,
second: the image visualised labels,
third: removing the black pixels predicted as part of symbol,
bottom: removing the black pixels predicted as part of shading)

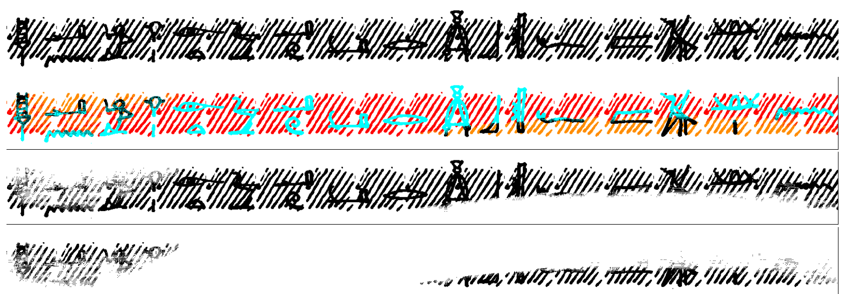


Fig 5.2.5: An example of 32x32 grid applying FFT
(top: raw,
second: the image visualised labels,
third: removing the black pixels predicted as part of symbol,
bottom: removing the black pixels predicted as part of shading)

Grid	Precision	Recall	F1-Score
6x6	0.9449	0.9449	0.9449
32x32	0.8785	0.8785	0.8785

Table 5.2.5: The accuracy of model with grid patterns applying FFT on test set

Grid	Precision	Recall	F1-Score
6x6	1	1	1
32x32	1	1	1

Table 5.2.6: The results model with grid patterns applying FFT regarding white label on test set

Grid	Precision	Recall	F1-Score
6x6	0.8294	0.8715	0.8499
32x32	0.6389	0.7386	0.6851

Table 5.2.7: The results of model with grid patterns applying FFT regarding shading label on test set

Grid	Precision	Recall	F1-Score
6x6	0.742	0.6733	0.706
32x32	0.3344	0.2393	0.2789

Table 5.2.8: The results of model with grid patterns applying FFT regarding symbol label on test set

5.2.2 Results of distance to the edge

In terms of grid 5x5 and 31x31, the results of with and without adding features with regard to separation of edge are in the table 5.2.9 - 5.2.12.

Grid	Separation of edge	Precision	Recall	F1-Score
5x5	No	0.9387	0.9387	0.9387
5x5	Yes	0.9383	0.9383	0.9383
31x31	No	0.8562	0.8562	0.8562
31x31	Yes	0.8538	0.8538	0.8538

Table 5.2.9: The accuracy of model with and without adding a feature regarding separation of edge on test set

Grid	Separation of edge	Precision	Recall	F1-Score
5x5	No	1	1	1
5x5	Yes	1	1	1
31x31	No	1	1	1
31x31	Yes	1	1	1

Table 5.2.10: The results of model with and without adding a feature regarding separation of edge regarding white label on test set

Grid	Separation of edge	Precision	Recall	F1-Score
5x5	No	0.7653	0.9484	0.8471
5x5	Yes	0.7653	0.9457	0.846
31x31	No	0.6167	0.5204	0.5645
31x31	Yes	0.6227	0.4649	0.5323

Table 5.2.11: The results of model with and without adding a feature regarding separation of edge regarding shading label on test set

Grid	Separation of edge	Precision	Recall	F1-Score
5x5	No	0.8334	0.47	0.601
5x5	Yes	0.8265	0.4715	0.6004
31x31	No	0.3196	0.4106	0.3595
31x31	Yes	0.333	0.4867	0.3954

Table 5.2.12: The results of model with and without adding a feature regarding separation of edge regarding symbol label on test set

5.3 Discussion

In the previous section, we indicated the results of several models and output images to demonstrate whether the approach can be dealt with the detection of the diagonal-line and remove shading This section discusses each model and room for improvements.

5.3.1 the number of grid

the number is odd

As displayed above, the case of 5x5, Fig 5.2.1 is the better performance than other cases. In terms of detecting diagonal lines, most pixels of these can be detected applied this model while some pixels are classified incorrectly. The black pixels on the edge of the symbol are prone to be mislabeled as part of shading because these black pixels are either ones that are both part of shading and part of symbol or close to the distance to the pixels of part of shading. Extending the size of grid for applying DFT can be obtained more information of pixels and we assume the performance is better than smaller grid. However, the larger the grid extends, the worse the performance is because the numbers of misclassified pixels increase with larger grid. Thus, as Fig 5.2.2 and Fig 5.2.3, the shading images based on the predicted labels include black pixels that belong to part of symbol and both 21x21 and 31x31 are not effective to remove diagonal lines compared to 5x5 grid. The evaluations of these grids are in the table 5.2.1- 5.2.4. As for accuracy, smaller grid is the best F1-score of the three. In the evaluation of specified label, white background is classified correctly in any grid. With regard to classifying part of shading, 5x5 is high F1-score of 85% while 21x21 and 31x31 are about 60%. The F1-score regarding shading label is low score overall. That is about 60%

in the 5x5, about 30% in the 21x21 and about 36% in the 31x31. In case of this approach, each pixel has information of itself and to raise resolution, we attempt to merge 4 pixels into one.

the number is even

As a result of enhancing resolution, it is more effective than before merging pixels. The output of 6x6 grid, Fig5.2.4 and the one of 32x32 is Fig5.2.5. Compared 5x5 grid, symbol image in the 6x6 grid is clearer to see. F1-score of accuracy also improves about 0.5%. F1-score of only symbol increases about 10% higher while F1-score of shading label improves slightly. On the other hand, in case of 32x32 grid, the F1-score increases 2% compared with 31x31, which is not merging pixels because of enhancing F1-score of classifying shading labels in particular. However, the evaluation of labelling as symbol gets worse and enlarge misclassified as part of shading. Thus, with regard to visualised images of 32x32, the output of this case is getting worse rather than 31x31 and the application of a large grid is not effective even if as to smaller grid, it is improvement to apply this approach.

5.3.2 the distance to the edge

In the result of 5x5, the black pixels of the boundary between part of shading and part of symbol are misclassified. Thus, regarding both part of shading and part of symbol, we classify labels and add a feature to separate these pixels into edge and inside respectively. Although we apply this approach into the grid of 5x5 and 31x31, F1-score is almost the same in both cases. Therefore, this approach is not effective.

5.4 Conclusion

In this chapter, results acquired by different grid patterns and approaches are proposed and discussed. As to white pixels on test set, we can completely classify as white pixels. The smaller grid has a contribution to detect diagonal lines and removing shading. In addition to that, merging pixels to enhance resolution improves the accuracy as to smaller grid. However, adding a feature for separating the pixels on the edge of both part of shading and part of symbol does not contribute to improve the accuracy. In case that the grid applying DFT/FFT is larger such as 21x21 and 31x31, it increases the number of misclassified pixels, and the accuracy is worse than that of smaller grid.

Chapter 6

Conclusion and Future Work

6.1 Summary

In order to promote the research of ancient Egyptian hieroglyphs, it is necessary to detect diagonal lines and remove them. In this project, we focus on detecting hatching and adopt the approaches of combination of both DFT/FFT method and SVM as the supervised learning approach. SVM parameters is default. Thus, we mainly evaluate the accuracy changing the number of grid applying DFT/FFT and adding another feature. As to dataset to learn the SVM model, we develop the tools to superimpose shading pieces into symbol blocks automatically. From the number of grid point of view, we propose some grid patterns for applying DFT/FFT. In terms of the pixels of white background, we can detect them completely in any grid size. The small size of grid are less misclassified pixels and bring the higher accuracy of detecting diagonal lines compared to larger grid. Merging pixels to enhance resolution under the condition of small size of grid decrease to the amount of misclassified pixels, label of part of shading in particular. Adding labels for detecting the pixels on the edge as a feature does not contribute to enhance the accuracy.

As mentioned above, we achieved the following primary objectives.

(1)Identify the hatching superimposing on the images of ancient Egyptian-hieroglyphs utilising FFT algorithm

(2)Create the dataset for machine learning and develop an tool to generate-dataset automatically

(3)Remove the hatching from the images of ancient Egyptian hieroglyphs-

based on the result of analysing FFT values
The best model in this project is the size of grid 6x6 and merging pixels when adding a feature. The accuracy is 94.49%.

6.2 Future Work

The combination of DFT/FFT method and SVM has accomplished the good performance and the introduction of DFT/FFT values as features brings to the contribution to enhance the accuracy of detecting shading. However, there is still rooms to solve tasks as the future research work.

6.2.1 Graphical User Interface

Developing tools to generate images save time to work on the process of detecting and removing shading. We develop tools to generate merging two images. However, the work cropping images from the hieroglyphs text uses the existing image processing software. Thus, developing GUI having the similar function such as the existing software promotes the efficiency of the process of removal task.

6.2.2 Parameter Tuning

In this project, the hyperparameters of SVM model is default. Thus, implementing grid search is desirable to find the optimal hyperparameters.

6.2.3 Labelling

As to the black pixels that are both part of shading and part of symbol, we classify these pixels as part of shading in this project. The objective is to detect diagonal lines, so regarding these pixels,labelled as part of shading is appropriate process. However, in the results of this approach, most pixels which is classified as part of symbol are labelled as part of shading incorrectly. Thus, not only shading, but also most symbols are output as predicted shading labels. It is because FFT values of amplitude and phase of both labels are almost the same values, and it is tendency to classify incorrectly under the condition that SVM parameters are default. Owing to deal with issue, it is desirable to consider the optimal hyperparameters of SVM model and threshold of FFT values to separate labels.

References

[1] K. Sethe, *Urkunden der 18.Dynastie.* J.C.Hinrichs'sche buchhandlung, 1927.

[2] S. Ghose, N. Singh, and P. Singh, "Image Denoising using Deep Learning: Convolutional Neural Network," in *2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pp. 511–517, 2020.

[3] A. K. Boyat and B. K. Joshi, "A Review Paper : Noise Models in Digital Image Processing," *Signal & Image Processing : An International Journal*, vol. 6, pp. 63–75, 4 2015.

[4] R. Kaur and M. Singh, "Stroke based online handwritten Gurmukhi character recognition," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 598–601, 2016.

[5] F. Alirezazadeh and M. R. Ahmadzadeh, "Effective staff line detection, restoration and removal approach for different quality of scanned handwritten music sheets," *Journal of Advanced Computer Science & Technology*, vol. 3, 6 2014.

[6] R. Komatsu and T. Gonsalves, "Removing noise from handwritten character images using U-Net through online learning," in *2018 International Conference on Image and Video Processing, and Artificial Intelligence* (R. Su, ed.), vol. 10836, pp. 95 – 99, International Society for Optics and Photonics, SPIE, 2018.

[7] A. Sun and X. Zhang, "Removing Grids from Digital Image with Handwritten Chinese Text Based on Cycle-Consistent Adversarial Networks,"

in *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*, pp. 246–251, 2020.

[8] M.-J. Nederhof, “OCR of handwritten transcriptions of Ancient Egyptian hieroglyphic text*,” in *Altertumswissenschaften in a Digital Age. Egyptology, Papyrology and beyond ;Proceedings of a conference and workshop in Leipzig*, 2015.

[9] P. P. Patro and C. S. Panda, “IJESRT INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY A REVIEW ON: NOISE MODEL IN DIGITAL IMAGE PROCESSING,” © *International Journal of Engineering Sciences & Research Technology*, vol. 5, pp. 891–897, 2016.

Appendices

User Manual

System and Package Requirements

The system requirement is CentOS 7 having NVidia GTX 1060 6GB. The version of Python is 3.8.10. The following table indicates the requirements for the packaged used in the program.

Package	Version
Ipython	7.25.0
Joblib	1.0.1
Notebook	6.4.0
Numpy	1.21.1
Opencv-contrib-python	4.5.3.56
Pandas	1.3.0
Scikit-learn	0.24.1
SVC	0.1

The requirements for packaged used in the program

Ethics Self-Assessment Form

UNIVERSITY OF ST ANDREWS
TEACHING AND RESEARCH ETHICS COMMITTEE (UTREC)
SCHOOL OF COMPUTER SCIENCE
PRELIMINARY ETHICS SELF-ASSESSMENT FORM

This Preliminary Ethics Self-Assessment Form is to be conducted by the researcher, and completed in conjunction with the Guidelines for Ethical Research Practice. All staff and students of the School of Computer Science must complete it prior to commencing research.

This Form will act as a formal record of your ethical considerations.
Tick one box

☐ **Staff Project**
☒ **Postgraduate Project**
☐ **Undergraduate Project**

Title of project

Removing noise from hand-written transcriptions

Name of researcher(s)

Junichi Hattori

Name of supervisor (for student research)

Mark-Jan Nederhof

OVERALL ASSESSMENT (to be signed after questions, overleaf, have been completed)

Self audit has been conducted **YES** ☒ **NO** ☐

There are no ethical issues raised by this project

Signature Student or Researcher

Junichi Hattori

Print Name

Junichi Hattori

Date

01/06/2021

Signature Lead Researcher or Supervisor

Print Name

Mark-Jan Nederhof

Date

02/06/2021

This form must be date stamped and held in the files of the Lead Researcher or Supervisor. If fieldwork is required, a copy must also be lodged with appropriate Risk Assessment forms. The School Ethics Committee will be responsible for monitoring assessments.

Computer Science Preliminary Ethics Self-Assessment Form

Research with human subjects

Does your research involve human subjects or have potential adverse consequences for human welfare and wellbeing?

YES ☐ **NO** ☒

If YES, full ethics review required
For example:
Will you be surveying, observing or interviewing human subjects?
Will you be analysing secondary data that could significantly affect human subjects?
Does your research have the potential to have a significant negative effect on people in the study area?

Potential physical or psychological harm, discomfort or stress

Are there any foreseeable risks to the researcher, or to any participants in this research?

YES ☐ **NO** ☒

If YES, full ethics review required
For example:
Is there any potential that there could be physical harm for anyone involved in the research?
Is there any potential for psychological harm, discomfort or stress for anyone involved in the research?

Conflicts of interest

Do any conflicts of interest arise?

YES ☐ **NO** ☒

If YES, full ethics review required
For example:
Might research objectivity be compromised by sponsorship?
Might any issues of intellectual property or roles in research be raised?

Funding

Is your research funded externally?

YES ☐ **NO** ☒

If YES, does the funder appear on the 'currently automatically approved' list on the UTREC website?

YES ☐ **NO** ☐

If NO, you will need to submit a Funding Approval Application as per instructions on the UTREC website.

Research with animals

Does your research involve the use of living animals?

YES ☐ **NO** ☒

If YES, your proposal must be referred to the University's Animal Welfare and Ethics Committee (AWEC)

University Teaching and Research Ethics Committee (UTREC) pages
http://www.st-andrews.ac.uk/utrec/