Stamp Verification

Final Year Project Final Report

A 4th Year Student Name

Ha Minh Nghia

Ha Long Duy

Instructor

Dr. Phan Duy Hung

Bachelor of Computer Science

Hoa Lac campus - FPT University

22 December 2021

Acknowledgement

We would like to thank our instructor, Dr. Phan Duy Hung for his patience and time, and for instructing and advising us enthusiastically.

We would like to thank all at my University, FPT, for giving us the best environment to study and grow over the years.

We would like to thank our classmates in CS1401, for letting us meet amazing people and learn a lot from them.

We always remember our family's encouragement and support. Thanks to them, we have the will, the energy and the confidence to pursue our goals.

Abstract

Stamps have become one of the most important security features in big companies where huge amounts of documents need processing everyday. The stamp attached to a document is used to determine the authenticity of that document so that it is necessary to identify whether a stamp is forged or genuine. However, because of the widespread use of high-quality color copiers, it is now possible to produce forged papers with forged stamp images that are easily mistaken for genuine stamps. This created a big risk for many companies in data security and documents authentication. Therefore, an automatic system for stamp verification needs to be developed to deal with this problem.

This thesis presents a practical approach for stamp verification, based on the 3 stages process similar to some previous work: Stamp segmentation, classification stamp or non-stamp and stamp authenticity verification. In each stage, this thesis tries and tests new algorithms/methods to give a new way of solving the problem in each stage. Firstly, in our approach, an unsupervised learning machine method is implemented to detect all the objects in the input image, so all the regions including stamps and text are extracted. Next, two separate models of Support Vector Machine classification are constructed. The first one is to distinguish between stamps and other objects in a document. The second model will determine the object which was classified as stamps in the first model whether it is genuine or not. The results show that this approach can perform the stamp verification tasks effectively.

Keywords: Stamps verification, image segmentation, Support Vector Machine

Table of contents

List of figures

1. Introduction

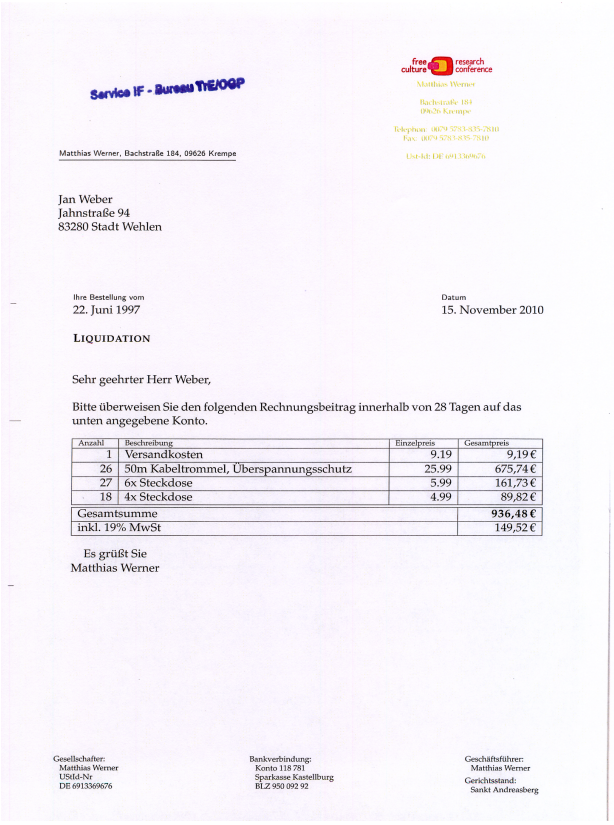
1.1. Overview

Nowadays, security is an indispensable aspect in the operation of big companies, especially in banks, insurance companies and financial companies. Every day, thousands of invoices, contracts, certificates and documents are handled. Therefore, it's necessary to have a way to guarantee the authenticity of the content. The stamp is one of the most widely used security methods, along with handwritten signatures. Every stamp on a document highlights the significance and purpose of the document. However, with the popularity of the internet these days and the availability of technology, it is easier for the general public to forge stamps. Big companies process a huge volume of documents everyday so that the risk of forged stamps is very big. Companies can lose hundreds of thousands or even millions of dollars just by errors caused by forged stamps. Moreover, forged stamps can affect the company's reputation and the customer's trust [1]. In some banks, stamp verification relies on manual handling work instead of using software [2]. Therefore, there are many cases where the forged document is accepted as genuine. In order to solve this problem, an automated stamp verification method is needed to be developed.

Stamps vary greatly in shape, size, and color from one company to the next, as well as within a single organization's departments. Stamps come in a variety of shapes, including textual, pictorial, regular (official), and irregular (for fun) as seen in Figure 1. Every stamp on a document highlights the significance and purpose of the document.

Figure 1. *Examples of stamps on documents.*

Figure 2 shows a sample of a basic document in the data set used throughout this thesis. Typically, a document contains from 2 to 3 stamps and the rest is text or text box. For some document stamps can overlap others components such as signatures, text, logos or in some cases, others stamps. There are many ways to forge stamps, yet the most simple and efficient technique is photocopy. In this technique, the document, for example an invoice, is scanned, digitally modified and printed again, or, alternatively, some parts are photocopied from another one.



Service IF - Bureau THEOGP

free culture research conference

Matthias Werner

Bachstraße 184
09626 Krempe

Telephone: 0879 5783-835-7810
Fax: 0879 5783-835-7810

Ust-Idt: DE 6913369676

Matthias Werner, Bachstraße 184, 09626 Krempe

Jan Weber
Jahnstraße 94
83280 Stadt Wehlen

Ihre Bestellung vom
22. Juni 1997

Datum
15. November 2010

LIQUIDATION

Sehr geehrter Herr Weber,

Bitte überweisen Sie den folgenden Rechnungsbeitrag innerhalb von 28 Tagen auf das unten angegebene Konto.

Anzahl	Beschreibung	Einzelpreis	Gesamtpreis
1	Versandkosten	9.19	9,19 €
26	50m Kabeltrommel, Überspannungsschutz	25.99	675,74 €
27	6x Steckdose	5.99	161,73 €
18	4x Steckdose	4.99	89,82 €
Gesamtsumme			**936,48 €**
inkl. 19% MwSt			149,52 €

Es grüßt Sie
Matthias Werner

Gesellschafter:
Matthias Werner
USIdd-Nr
DE 6913369676

Bankverbindung:
Konto 118 781
Sparkasse Kastellburg
BLZ 950 092 92

Geschäftsführer:
Matthias Werner
Gerichtsstand:
Sankt Andreasberg

Figure 2. *A sample document that contains stamps*

In this thesis, the process of stamp verification inherits the process by many previous works which divide it into three stages. This thesis follows this 3 stages process closely but for each stage, the way of implementation is different with different methods and algorithms being applied. The goal is to present a new practical method for this stamp verification problem. First, the stamp must be identified and extracted from the document image. By using color space transformations and k-means clustering, the scanned color image is split according to colors and then used the XY-cut algorithm, all the components in the image are segmented into candidates which contain both stamps candidates and other elements like logos, text, etc. The second stage is to classify the extracted candidates to identify which candidates are stamps, which are not stamp. Each candidate image is passed through a Support Vector Machine (SVM) model to identify whether that candidate is a stamp or not. After getting all the stamps, in the final stage, another SVM model will handle the

task of classifying whether the stamps are genuine or forged.

1.2. Problem and Challenging

1.2.1. Problem statement

Several approaches for document source authentication that leverage intrinsic features (properties that originated in a regular document generation process) have been published in the literature. These methods work by comparing an arriving ("questioned") document to legitimate documents from the same source that already exist in the database. If the features are significantly different, the arriving document is regarded as suspicious, and an expert examination can be requested.

Extrinsic features (properties added only to assist document authentication), such as signatures, counterfeit protection system (CPS) codes, and stamps, can be evaluated in addition to intrinsic features. Automatic techniques for signatures and CPS codes have been presented in the literature. To the best of our knowledge, no automatic method for validating the genuineness of stamps has yet been presented.

To check the authenticity of a stamp, the stamp must first be retrieved from a document image. The challenge is how a stamp can be recognized automatically and how it differentiates from other elements in the document image, such as logos. Stamps have previously been detected using certain placements, forms, and colors. However, when it comes to papers having pictorial content, such as logos or other graphics, these attributes will not be enough to ensure accurate stamp detection. Other aspects, such as the extremely precise manner of imprinting a stamp, must be included to adequately characterize the essence of a stamp imprint. It becomes evident that stamp detection and stamp verification are two challenges that are closely related and overlap in several ways.

1.2.2. Challenges

One of the most critical and well-known issues in stamp authentication is the considerable intra-class diversity of stamps, which means that stamps, in general, do not have a template. It's a partially graphical and partially textual object that can go anywhere in the document. The differences can be found in the shape and color of the stamp, as well as the print quality and rotation, and even two imprints of the same stamp can appear to be very different. Moreover, stamp imprints on the document can be overlapped by others components as described above, this makes the task of stamp segmentation much more difficult. For example, when a stamp is overlapped by a logo, there is a big chance that these two entities are different in color. In that case, we can detect the stamp based on its distinctive color. In the case when the stamp and the logo are the same color, the challenge is bigger because the approach based on color is no longer usable. Now, we have to include the shape of stamp to distinguish, this make the process longer and more complex.

The presence of particle knowledge during training is the second challenge while training an automatic stamps verification system. During the training phase, we simply need to access genuine stamps from the company's security system in the realistic scenario. However, during operations, we want the system to be able to accept authentic documents as well as reject forgeries. To address this, each company's forging stamps should be necessary for a more accurate classification. However, requiring companies to furnish stamps that they falsify is not reasonable in general. Even if they can be collected and manufactured from the service provider or a third party, it would be difficult to create acceptable counterfeit data that is completely different from real stamps.

Thirdly, the amount of training data is always taken into account. Companies are frequently asked to provide simply a few samples of their stamps during the security process in the genuine application. Meanwhile, other methodologies require a sufficient amount of data to be effective. As a result, creating an effective system with the minimal data collected from organizations and enterprises in a real-world scenario is difficult. Even if a significant number of companies submit their stamps to the system during the training phase, the classifier's performance needs to be very good for new companies that supply only a few stamps.

Finally, an effective real-world stamps verification system, like the face recognition system, should be able to deal with the one-shot learning problem. That is, the verification application should be able to distinguish between genuine and fraudulent stamps based on only one genuine stamp. Deep learning algorithms, on the other hand, do not perform well with a small dataset, especially if only one training example is available.

1.3. Related works

Unlike handwritten signatures, stamp verification is quite limited in terms of number of published works. Many previous works only focus on the task of detecting and segmenting stamps from a document. The majority of these works applied many different color-based [3] shaped-based features [4] or used geometric features with keypoint descriptors to detect and segment stamps [5]. Their ultimate goal is to separate stamps from logos, text, and other information in original document images. Chen et al. [6] developed a method to detect stamps on checks of Chinese banks with a region-growing algorithm. Micenková el at [7] presented a new method for detecting and extracting stamps of various colors, even stamps that overlap with a signature or text of a different color. There are two steps in this work, first is to separate every part of the image from text and background with color clustering, then all the candidates obtained are classified to be whether stamp or not by using a set of features. Figure 3 shows the result of this work is very positive. Even though the stamp is partly overlapped by a logo, this approach still manages to detect and segment the stamp very well. Figure 4 shows the sample result with the detected stamp is marked by

green boxes.

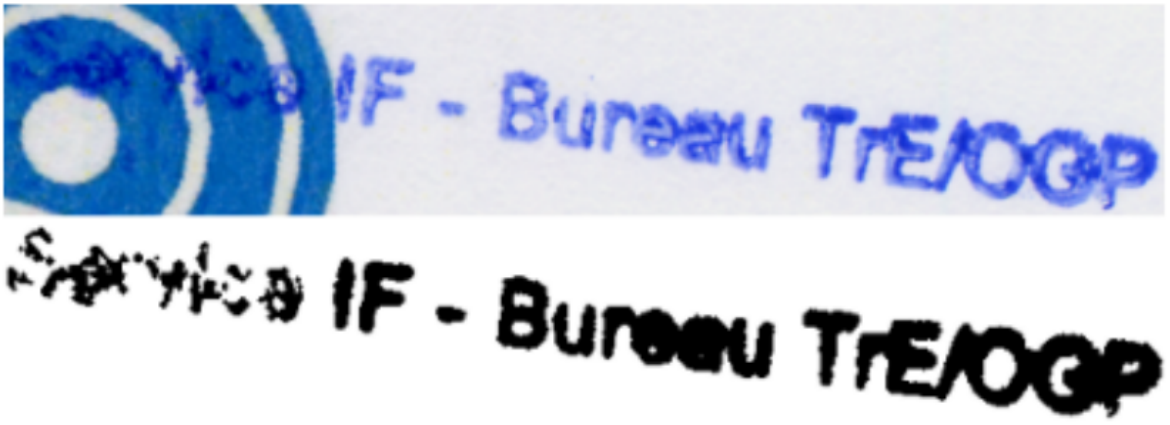


Figure 3. *The result of a correctly detected and segmented stamp*

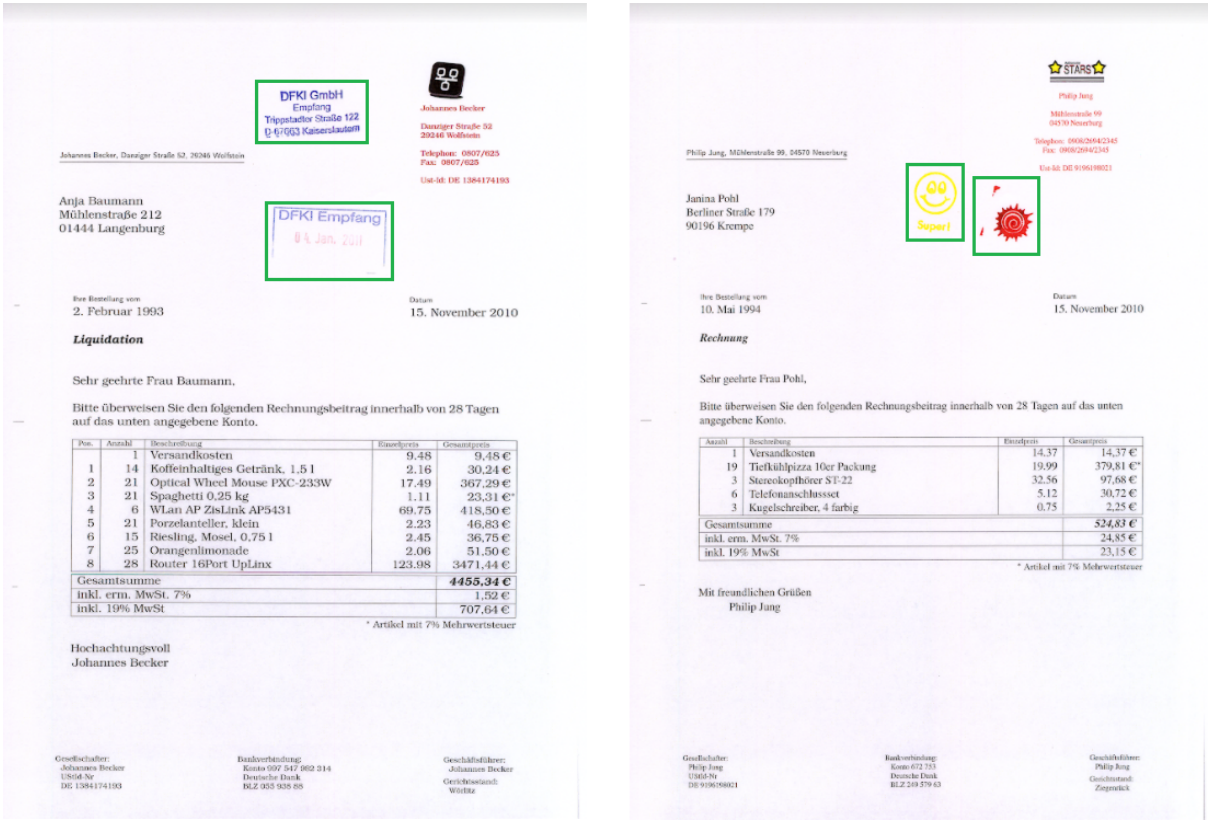


Figure 4. *Sample stamps detected marked in green boxes*

In many works which are dedicated to stamp detection and segmentation, the work that is worth paying attention to is the work by Younas et al [8] with the name "D-StaR: A Generic Method for Stamp Segmentation from Document Images". They proposed a brand-new approach called D-StaR based on deep learning, capable of

handling stamp segmentation in any color, shape, size, orientation and this approach can even detect overlapping stamps. They used Fully Convolutional networks (FCN) to segment stamp masks from scanned document images. Moreover, for pixel-based evaluation and reforming the original stamps, contour refinement is performed to the expected masks. The detailed architecture of D-StaR is shown in Figure 5.

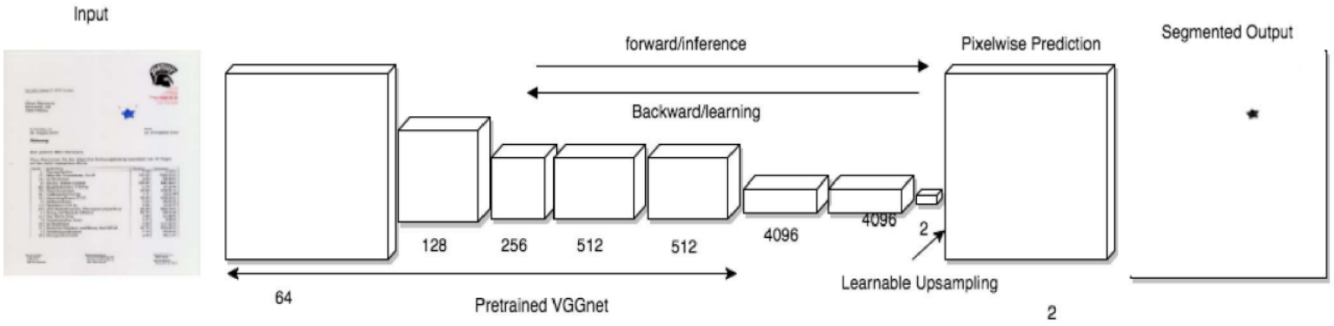


Figure 5. *D-StaR architecture with input image and pixel-level segmentation result of FCN*

D-Star is the first method to use deep learning to segment stamps and get amazing results. It is also the basis for other studies such as [7], [9], [10]. Many other previous works solved the stamp verification entirely, both stamp segmentation and stamp authentication verification. Chung et al [11] focused on a data set of Chinese antique seal/stamp, in the stamp segmentation step they proposed a method based on geometric transformation and geometric transformation to find borders and align the perspective for two imprints. Then to verify the authenticity of stamps, they calculated the similarity by PSNR and SSIM indexes as the detection metrics. Chung et al improved their method in the next work [12]. The work by Takahiko Horiuchi [13] brings up a problem that many techniques for seal/stamp verification only solve this problem as a general pattern matching problem, therefore the paper presented a new approach to deal with this problem. In the experiment step, an interesting way to test the method is used, the paper used both binary reference images and 3D reference to test the accuracy of the method. Works [14-17] presented some different ways in the stamp verification problem which include: using edge difference histogram, neural network, based on Average Relative Error, judging the percent of difference inside and difference outside to determine whether a stamp is genuine or not.

Finally, there are 2 works that inspired us to develop 3 stages of stamp verification in this thesis, all by the author Micenkov et al. The first work [10] introduced three main stages of the problem. First, they segmented all every component in the original document image. Then, they used SVM to classify whether that component is stamp or not. Finally, they used features like color, shape, and print to verify stamps. With the same process, the work by Micenková el at [9] inherited two of their previous work [7], [10] and their work is improved by using k-means clustering additionally in the segmentation stage, they still used features to classify genuine and forgered stamps. In this study, we try and test new algorithms/methods to propose a best way

for solving the problem in each stage.

1.4. Contribution

In this thesis, we developed an efficient approach to the stamp verification task. Our approach is divided into 3 stages: Stamp segmentation, classification stamp or non-stamp, stamp authentication verification. Each stage we try methods, algorithms to develop our own method to implement that stage.

Concretely, in the first stage - stamp segmentation - we use color space transformation and k-means clustering to detect every component on a document including text, logo, signatures and stamp. Then the XY-cut algorithm is used to segment all the detected components. In the second stage, we simply use a SVM model to classify the segmented components in the last stage to determine which is stamp and which is not. We try to make the methods and algorithms used in the first two stages simple and easy to implement in code and also give a good result in each stage. All the methods in these two stages are basic, easy to understand and easy to implement.

In the final stage, the task is to classify which stamp is genuine, which is forged. Features extraction is applied and another SVM model is implemented. In this stage, we test various feature extractors from the scikit image library to select the one that produces the best result. Then combine the selected features extractor with some classification model. This process covers many methods and model to get the best result possible.

1.5. Outline

In thesis, we address the problem of stamp verification, specifically:

Section 1 gives a gentle introduction about the problem, motivation and related works about stamp verification.

Section 2 gives an overview about the data set which is used throughout this work and explains the preprocessing process implemented.

Section 3 demonstrated our approach to effectively solve three tasks described above: stamp detection and segmentation, classify stamp or non-stamp, stamp verification. In each stage, we will explain and clarify all the models and methods implemented in that stage.

Section 4 showed the evaluation method and some experiment results between different methods on different stamp verification.

Section 5 concludes the thesis and then makes some future work in the subject.

The final section is the list of all reference works helping to create this thesis.

2. Preprocessing and methodology

2.1. Data and preprocessing

2.1.1 Dataset overview

This thesis evaluated both classification stages, detection and verification, separately with the public data set StaVer. The published data set contains 400 document pages for evaluation of stamp detection algorithms. This data set contains stamped invoices with color logos and texts. Stamps are often overlapped with text or other objects. It was generated by printing automatically generated invoices, stamping them manually and scanning them in color with a resolution of 600dpi. To limit the effort of ground-truth labeling, the lower resolutions were obtained by downscaling. To evaluate the performance of verification, we created and published a new data set4 of copied documents. A total of 14 invoices were selected randomly from our data set and their copies were made on 5 different models of Ricoh Aficio copy machines. We obtained 70 images with 78 copied stamps altogether.

The reason we choose this dataset is that they are public, free to access, and contain skilled forgeries.

The data set is available at http://madm.dfki.de/downloads-ds-staver.

2.1.2. Preprocessing

First, we can see that color factor is not important when comparing the dissimilarity of genuine and fraudulent stamps. That is why we convert all scanned documents in the data set to grayscale.

As mentioned earlier, our dataset contains salt pepper noise in most signature images. However, the noise density in each image is quite sparse. By using a Median filter with kernel size 3x3, most salt and pepper noise are easily eliminated. Due to its good performance for particular specific noise types such as "Gaussian," "random," and "salt and pepper" sounds, the median filter is one of the most well-known order-statistic filters. The median filter replaces the center pixel in a MxM neighborhood with the median value of the corresponding window. It's worth noting that noise pixels are thought to be significantly different from the median. A median filter, which is based on this concept, can eliminate this type of noise problem. This filter is used to reduce noisy pixels from protein crystal pictures prior to binarization. The illustration of the Median filter is shown in figure 6.

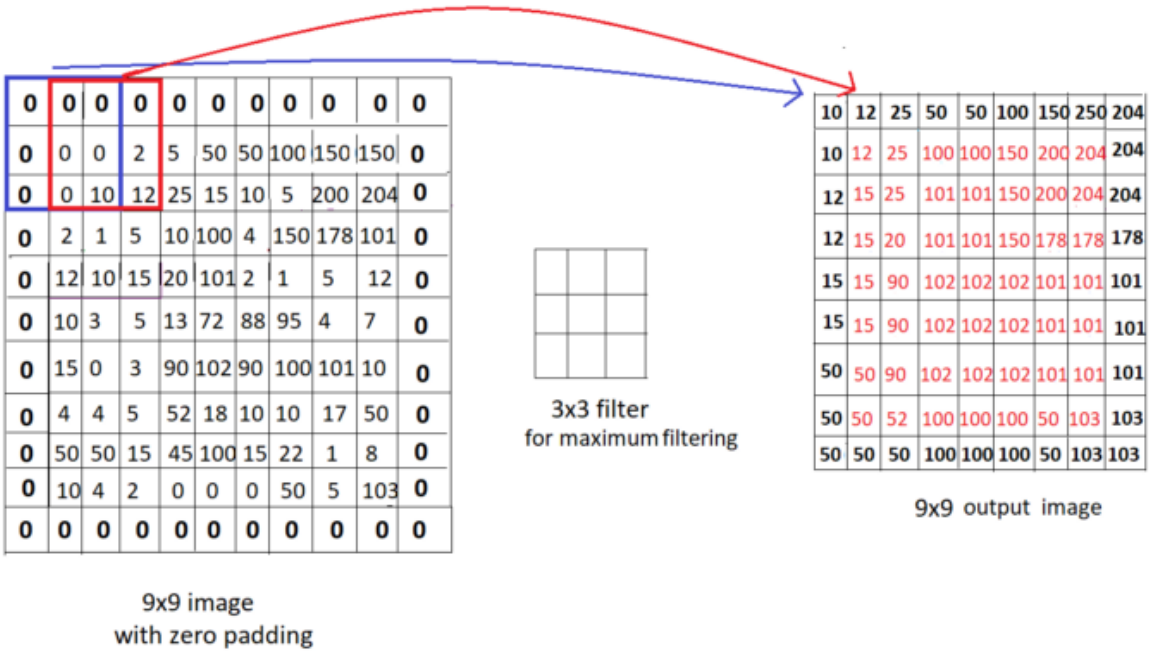


Figure 6. *Median filter illustration*

After removing noise, we use Otsu threshold method to convert the gray images to be the black and white (only including the pixel value of 0 or 255). In addition, we invert the image to make the background become zero and only the signature's area has pixel value 255.

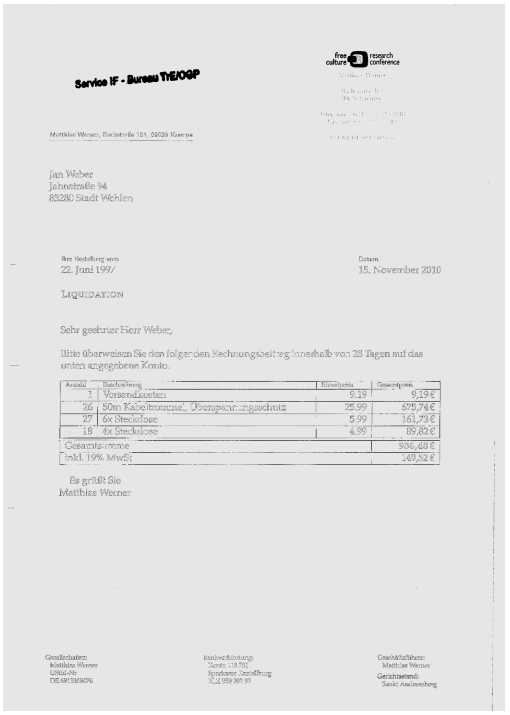


Figure 7. *Grayscale processing result*

Finally, we resize all stamps images to a fixed size for feeding into the network. With the stamps of StaVer, we decide to use the shape 128x128 as the work on. However, most images do not have square size, and some of them have a very long width compared to the height. To ensure that the overall shape of the signature in the image is not deformed after resizing, we add the zero-padding (black padding) to change the shape of the rectangular image into square shape before resizing it into the shape 128x128 like the other dataset. The sample result is shown in figure 7 and 8.

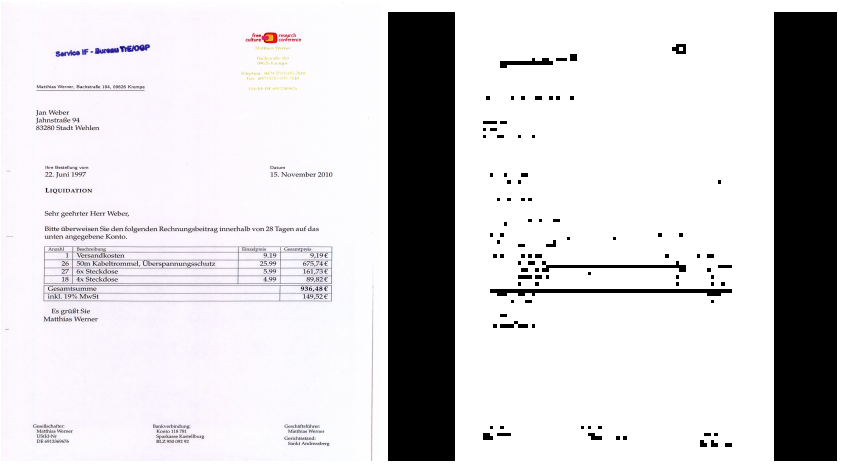


Figure 8. *Original picture and after preprocessing picture*

2.2. Methodology

In this thesis, the hardcopy of the document is scanned in color and the image is segmented completely. Candidates for solving the stamp identification problem are identified as rectangular segments, from which features are extracted. Candidates are classified as stamps or non-stamps using a binary classification system (logos, text etc.). Segments designated as stamps are further categorized to distinguish between legitimate and fake stamps, the process is shown in figure 9. Stamps are considered single-color (blue, green, red, etc.) items in this study. As a result, the primary principle behind detecting them in a picture is to group components that are the same color and are close to one another. Special examples of multiple-color stamps are detected as several stamps, which are subsequently combined.

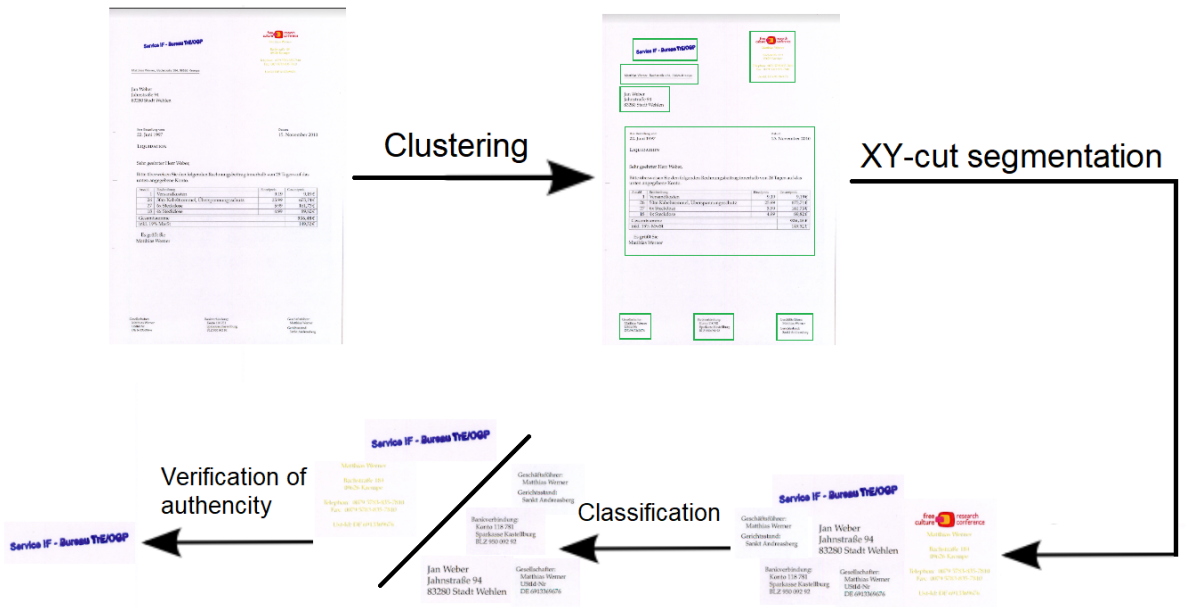


Figure 9. *Stages of stamps detection and verification*

This thesis identifies candidate segments by using this principle. One candidate segment is formed by each stamp in the image. If the image contains color logos or pictures, these (or single-color parts of them) are also identified as candidate segments. Then determine which parts match to genuine stamps and which match to printed objects such as duplicate stamps or logos. The following section will give detail of this approach.

2.2.1. Segmentation

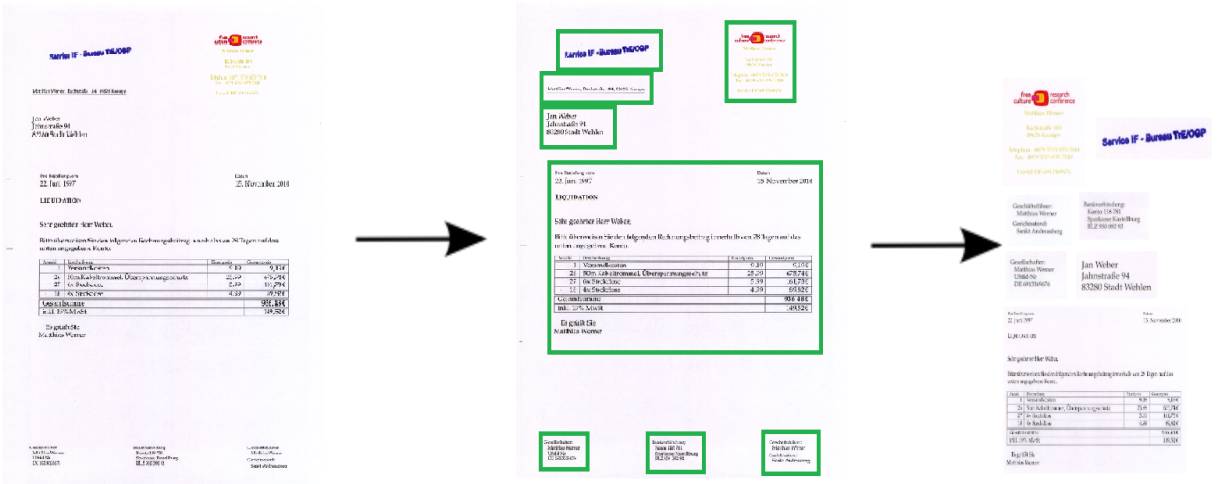


Figure 10. *Segmentation stages*

In the segmentation stage, the goal is to detect all the components in the original document image and crop each of those components into a new image. These new

images will serve as a new material for the next classification stages. Because of the significant correlation among the channels, the RGB color model is not suitable for image segmentation. In order to work with color stamps, separate the backdrop, black text, and other roughly achromatic (white, black, and grey) components of the image first so the image is converted from an RGB image into a grayscale image. After that, Otsu's Binarization [18] with a threshold value of 200 is applied to split the image into background and object. The image is also blurred with Gaussian blur pixels of similar features are desired to be assigned the same label, spatially continuous pixels are desired to be assigned the same label, the number of unique labels is desired to be large. With the image processed and ready to go, the next step is to find every cluster in the image, once again the Otsu's Binarization [18] is applied along with k-mean clustering is also applied. For each cluster, Canny Edge Detection [19] of OpenCv is implemented to detect every edge on that cluster. Also, more OpenCv operations are applied including: cv2.dilate to make the object thicker, morphologyEx to close the cluster to the object and then finding the contour can be easy. Last, the largest contour is selected then crop each object detected and save it as a new image and move it to a different folder.

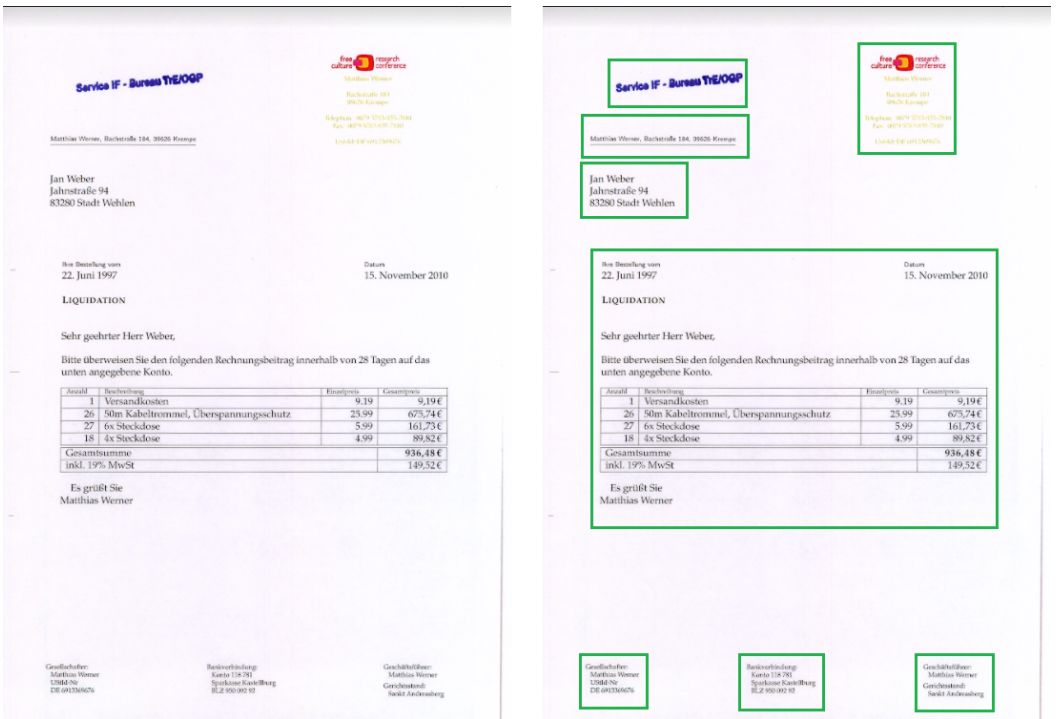


Figure 11. *Sample document image with candidates marked in green boxes.*

In order to segment every candidate properly, the XY-cut algorithm [20] is used to recursively partition the page into rectangles, resulting in candidate solutions with the smallest bounding boxes. The sample result is shown in figure 11. In conclusion, the method in the first stage we introduced above has 4 steps: (1) Preprocess image by resize, binarize, bur; (2) Find cluster; (3) For each cluster, detect edges, close figures,

find contours; (4) Select largest contours. This method can be considered to be a good and simple way to deal with stamp segmentation problem.

2.2.2. SVM classification: Stamp/non-stamp

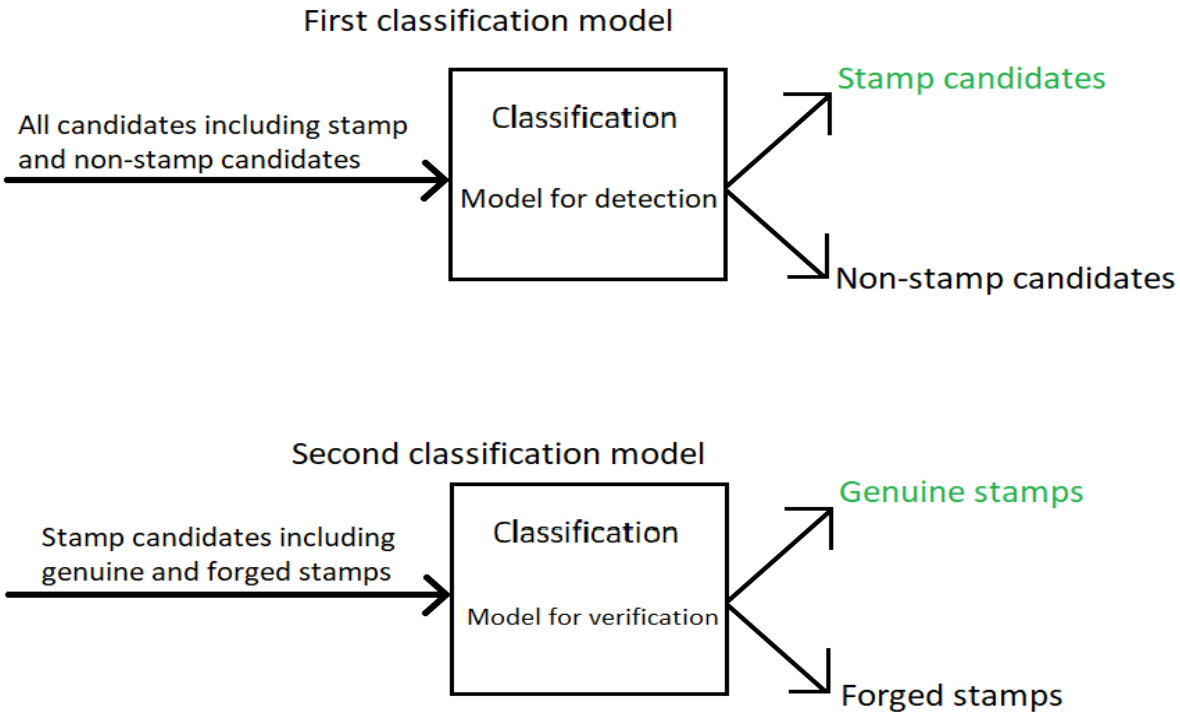


Figure 12. *Two models for detection and verification.*

Now all the candidates are segmented from the last step, those candidates contain all kinds of non-stamp candidates such as logos, parts of picture or even forged stamps. First, the task is to classify whether that candidate is stamp or not and finally with all the candidates which were identified as stamp, another classification is needed to determine that stamp candidate is genuine or fraudulent. This process is clarified in figure 12.

In this thesis, we decided to use Support Vector Machine for both classification models in each of two stages presented above. SVM is an excellent classification algorithm. It is a supervised learning algorithm that is primarily used to categorize data. SVM learns from labeled data. The main advantage of SVM is that it can be used for classification as well as regression problems. To separate or classify two classes, SVM draws a decision boundary, which is a hyperplane (A hyperplane is a decision plane that separates between a set of objects having different class memberships) between them. Our problem is just a binary classification problem and SVM is perfect for this task but furthermore, SVM can be used for multiclass classification problems. The main objective is to segregate the given dataset in the best possible way. The distance between the either nearest points is known as the margin. The objective is to select a hyperplane with the maximum possible margin between support vectors in the

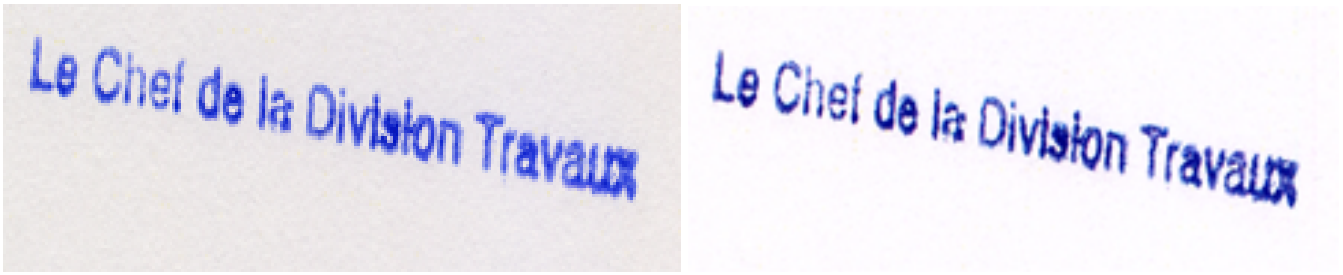
given dataset. SVM searches for the maximum marginal hyperplane in the following steps:

- Generate hyperplanes that segregate the classes in the best way. Left-hand side figure showing three hyperplanes black, blue, and orange. Here, the blue and orange have higher classification errors, but the black is separating the two classes correctly.
- Select the right hyperplane with the maximum segregation from the nearest data points as shown in the right-hand side figure.

In our compiling SVM process, we used hinge as a loss function. For l referring to the loss of any given instance, y[i] and x[i] referring to the ith instance in the training set and b referring to the bias term, the hinge loss can be computed by the formula:

$$l = max(0, 1 - y^{i}(x^{i} - b))$$

2.2.3. SVM classification: Genuine stamp/forged stamp



(a) Genuine stamp (b) Forged stamp

Figure 13. *Sample genuine and copied stamp after segmentation.*

Our final stage, also the most important stage, the task is to distinguish between genuine and forged stamps. In figure 13 is a sample of genuine and forged stamp, the difference can be recognized in hue and sharpness. Therefore, feature extraction is a suitable method in this case. This thesis uses several feature extractors from the scikit image library for feature extraction. These extractors help to create feature vectors, which are then fed into the SVM classifier. The purpose is to test various scikit image feature extractor and select the ones that produce the best results. The following is the list of extractors used in this work:

- Binary robust independent elementary features (BRIEF): An efficient feature point descriptor very fast both to build and to match. BRIEF easily outperforms other fast descriptors such as SURF and SIFT in terms of speed and terms of recognition rate in many cases. Intensity comparisons are performed for a randomly distributed number N of pixel-pairs for each keypoint, resulting in a

binary descriptor of length N. In comparison to the L2 norm, the Hamming distance can be employed for feature matching with binary descriptors, resulting in cheaper computing costs (scikit).

- Oriented FAST and rotated BRIEF (ORB): A fast and reliable local feature detector for computer vision tasks such as object detection and 3D reconstruction. It's built around the FAST keypoint detector and a tweaked version of the BRIEF visual descriptor (Binary Robust Independent Elementary Features). Its goal is to provide a quick and effective replacement for SIFT.
- Center surround extremas for realtime feature detection and matching (CENSURE): A set of scale-invariant center-surround detectors that outperform the competition, have superior computational properties than other scale-space detectors, and can be implemented in real time.
- Histogram of the grey scale image.
- Histogram of oriented gradients (HOG): A feature descriptor for object detection in computer vision and image processing. The technique counts the number of times a gradient orientation appears in a certain area of an image. Edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts are all comparable methods, but this one differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for enhanced accuracy.
- Corner peaks: Find corners in corner measure response image.

For stamps, there are many different features such as hue, uniformity, shape, etc. In many cases, using a certain features extractor doesn't help to improve or even make classification performance worse. Therefore, it is necessary to determine which features extractors to combine to achieve the best result. A process needs to be developed to find out which features should be utilized. A sample of 100 images is extracted from the training set and then fed into the SVM model to correctly estimate which features reduce model accuracy. The accuracy is returned and the features list with the highest accuracy is chosen. It was discovered that including the HOG and grayscale histogram extractors always produced more positive results, which is why they are used automatically to extract their respective features and feed them into the model. For the classifiers, the following models are chosen and implemented: Logistic Regression, SVM adj, SVM linear, SVC, K-nearest neighbors, Decision Tree, Random forest, AdaBoost, Naive Bayes, Gradient Boosting, Latent Dirichlet allocation. The feature vectors obtained from extractors are used as the input for these classifiers.

In the SVM classifier, it is necessary to choose two parameters C and gamma, these parameters have a big influence on the resulting accuracy. For this problem, a SkLearn machine learning library Method called “GridSearchCV” [21] which can compute the best possible parameter is applied. With C, the value is determined from 1.0e-02 to 1.0e10. With gamma, the value is determined from 1.0e-09 to 1.0e3. This helps to increase the accuracy significantly compared to the default parameter C and gamma.

This testing process to find the best performing feature extractor is an improvement in this thesis. It helps to use feature extractor and classification algorithms and get the best performance in stamp verification possible.

3. Experiments and conclusion

3.1. Evaluation and result

3.1.1. Evaluation

Evaluation is performed separately for stages: detection, classification, and verification. The public data set of 400 document pages is used and available at http://madm.dfki.de/downloads-ds-staver. Stamped invoices with color logos and text are included in this data set. Stamps are frequently layered with text or other objects. It was created by printing automatically generated invoices, manually stamping them, and scanning them in color at 600dpi. Lower resolutions were obtained by downscaling to reduce the effort of ground-truth labeling.

To evaluate the performance of verification, the data set is split into training set and test set with the ratio 8:2. All of the models in this thesis are trained and tested on these two sets.

3.1.2. Result

The results obtained from two classification models are very positive. Figure 14 shows the graph of training, validation accuracy compared to training, validation loss. We can see that the accuracy is alway above 0.8 while the highest loss is about 0.4, most of the time the loss is in range from about 0.2 to 0.3. The accuracy stays in range about 0.9 in the majority of the process. The graph show that our approach give a very good result with a simple way of implementing many basic methods and algorithms. In figure 15 is the graph of SVM model performance on the training set of classification stamp/non-stamp.

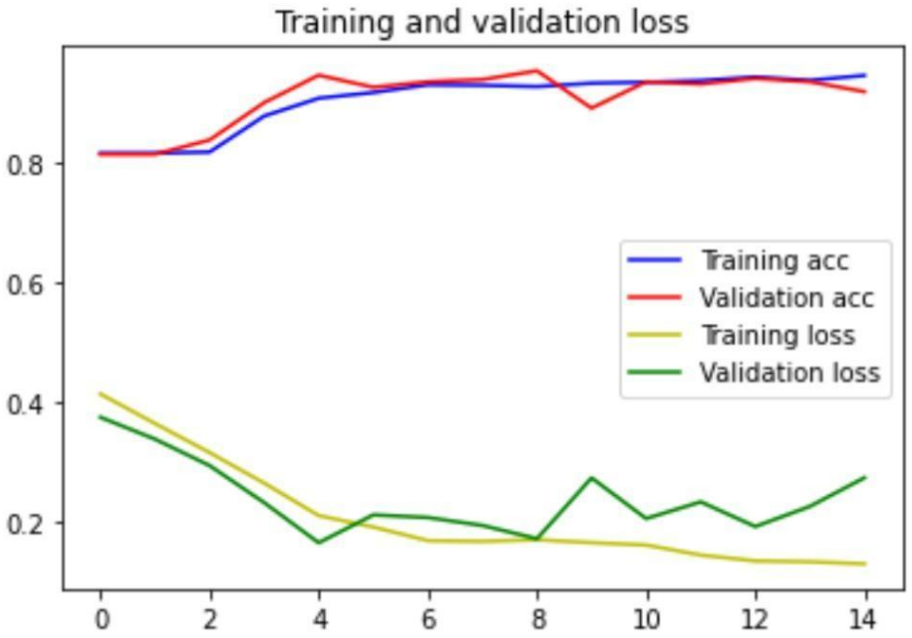


Figure 14. *Performance of first SVM model on training set.*

For the first SVM model, classify stamp or non-stamp, after training on a full data set, the mean accuracy is 0.9. The training pairs for the classification are formed after we get the embedding of all signatures using the train triplet net. Then we create the pairs, label them (see Table 8) and feed them to the classifier.

For the second classification, several extractors are used and tested to find the best combination of features among every possible combination of four extractors: ORB, Corner peaks, BRIEF and CENSURE. As presented in the previous section, histogram of gray scale image and HOG always produced better results so that always have to include them. After implementing many classify algorithms, the best result of the mean accuracy is 0.96 as shown in Table 1.

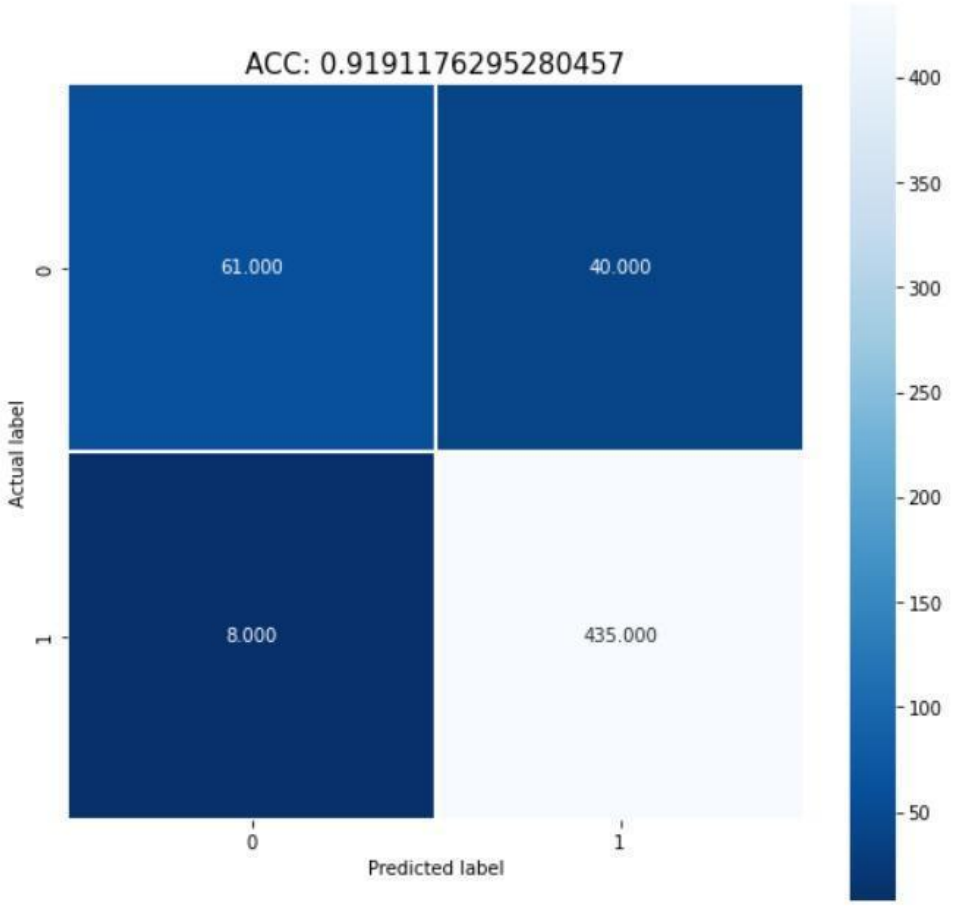


Figure 15. *Confusion matrix of first SVM model on test set.*

The results show that this approach can be considered to be a good option when it comes to stamp verification problems. The process of determining which is the best way to utilize features extractor is a good improvement and can be applied in many other classification problems.

Table 1. Result by classifiers

Classifier name	Accuracy
Logistic Regression	0.93
SVM, adj	0.82
SVM, linear	0.94

SVC	0.96
K-nearest neighbors	0.89
Decision Tree	0.91
Random Forest	0.97
Random Forest 2	0.82
AdaBoost	0.93
Naive Bayes	0.84
Gradient Boosting	0.93
Latent Dirichlet allocation	0.94

3.2. Conclusion and future work

3.2.1. Conclusion

This thesis introduces a simple and efficient approach to the stamp verification problem. We choose the process of 3 stages, the same as many previous works, including: Segmentation, classification stamp or non-stamp, classification forged stamp or genuine stamp. In each stage, we try, test and recommend the best model for the process.

In the first two stages, we apply basic and easy to implement methods and models such as: color space transformation, k-mean clustering, Support Vector Machine classification model. The final stage is where we apply and compare various methods and algorithms to come up with the best combination of features extractors and classification models. This stage gives us a better look about which models and algorithms should be prioritized in the process of determining the stamp's authenticity. All models and algorithms tested are popular, no need to introduce and can be implemented immediately. Hence, this approach can be considered a simple and effective way to handle stamp verification problems. Other works can apply this

approach, especially the works involving image segmentation or image classification.

3.2.2. Future work

In future work, we would like to develop this approach in more kinds of stamps. Right now, this thesis mainly focuses on stamps from the public data set that consist of entire stamps from foreign countries. The idea is to shift the focus to Vietnamese stamps, which contain Vietnamese letter patterns, symbols on them like the example shown in figure 16. This would make it easier to apply stamp verification methods for Vietnamese companies and organizations.

The data set in this work relies entirely on the published data set StarVer. Although this data set is large and gives good results when applied in our approach, we still don't have full control of data, some documents in this data set don't satisfy us. Therefore, in the future we would like to construct and publish our own stamp verification data set. With the intention to focus on Vietnamese stamps as said above, it's reasonable to construct a data set containing only Vietnamese stamps with distinctive Vietnamese patterns and symbols. This data set will fit with our approach to Vietnamese stamp verification that we would love to develop.

Figure 16. *Examples of Vietnamese stamps with distinctive patterns, symbols.*

Reference

1. SBA Stone Forest.: Mitigating Risks Associated with Seals. Stone Forest Business Advisors. https://www.stoneforest.com.sg/business-advisors/Articles/Article/mitigating-risks-associated-with-seals (2019)

2. Gao, W., Dong, S., Chen, X.: A system for automatic Chinese seal imprint verification. In: Proceedings of 3rd International Conference on: Document Analysis and Recognition, Montreal, Que., vol. 2, pp.660–664 (1995)

3. Micenkov, B., Beusekom, J. V.: Stamp detection in color document images. ICDAR, pp. 1125–1129 (2011)

4. Bhalgat, Y., Kulkarni, M., Karande, S., Lodha, S.: Stamp processing with examplar features. arXiv:1609.05001 (2016)

5. Forczmanski, P., Markiewicz, A.: Low-level image features for stamps detection and classification. CORES 2013 pp. 383–392 (2013)

6. Chen, L., Liu, T., Chen, J., Zhu, J., Deng, J., Ma, S.: Location algorithm for seal imprints on Chinese bank-checks based on region growing. Optoelectronics Letters, vol. 2, pp. 155–157, (2006)

7. Michenkova, B., Beusekom, J.V.: Stamp Detection in Color Document Images. Document Analysis and Recognition (ICDAR) (2011). https://doi.org/10.1109/ICDAR.2011.227

8. Younas, J., Afzai, M. Z, Malik, M. I., Shaifat, F., Lukowícs, P. & Ahmed, S.: D-StaR: A Generic Method for Stamp Segmentation from Document Images. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (2017). https://doi.org/10.1109/ICDAR.2017.49

9. Michenkova, B., Beusekom, J. V., Shafait, F.: Stamp Verification for Automated Document Authentication. Computational Forensics pp 117-129 (2015). https://doi.org/10.1007/978-3-319-20125-2_11

10. Michenkova, B.: Verification of Authenticity of Stamps in Documents (2012). http://acmbulletin.fiit.stuba.sk/vol3num4/micenkovaSPY.pdf

11. Chung, W. H., Wu, M. E., Ueng, Y. L., Su, Y. H.: Forged seal imprint identification based on regression analysis on imprint borders and metrics comparisons. IEEE Conference on Dependable and Secure Computing (DSC) (2018)

12. Chung, W. H., Wu, M. E., Ueng, Y. L., Su, Y. H.: Seal imprint verification via

feature analysis and classifications (2019). https://doi.org/10.1016/j.future.2019.04.027

13. Horiuchi, T.: Automatic Seal Verification by Evaluating Positive Cost. Proceedings of Sixth International Conference on Document Analysis and Recognition (2002)

14. Jin, H., Hao, Z., Tiegen, L.: Seal imprint verification using edge difference histogram. Proceedings Volume 8558, Optoelectronic Imaging and Multimedia Technology II; 855804 (2012)

15. Hong-yu, J., Zhen, G.: Research of the Seal Imprint Time Verification Method based on Neural Network. International Conference on Intelligent System Design and Engineering Application (2010)

16. Liang, J., Tong, X., Yuan, Z.: The Circular Seal Identification Method Based on Average Relative Error. Applied Mechanics and Materials, volumes 513-517 (2014)

17. Shuang, W., Tiegen, L.: Research on registration algorithm for check seal verification. Proceedings Volume 6833, Electronic Imaging and Multimedia Technology V; 68330Y (2007)

18. Yousefi, J.: Image Binarization using Otsu Thresholding Algorithm. (2015). https://doi.org/10.13140/RG.2.1.4758.9284

19. Canny, J., F.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8(6):679 - 698 (1986)

20. Sutheebanjard, P., Premchaiswadi, W.: A modified recursive X-Y cut algorithm for solving block ordering problems. Conference: Computer Engineering and Technology (ICCET), 2010 2nd International Conference on Volume: 3 (2010)

21. Ranjan, G., S., K., Verma, A., K., Sudha, R.: K-Nearest Neighbors and Grid Search CV Based Real Time Fault Monitoring System for Industries. IEEE 5th International Conference for Convergence in Technology (I2CT) (2019)

Appendix

Demo code

In this appendix, demo code is demonstrated for the approach presented in this work.

```
def processImg(that_img , img_size=None):
  DataSet = []
  LabelSet = []
  lengthV = []
  resList = []
  boolList = []
  pos = 0
  ind = 0
  useList = [True, True, True, True]
  #initialize feature detectors/extractors
  #Censure extractor
  detector = CENSURE()
  #ORB extractor
  detector2 = ORB(n_keypoints=50)

  featureVector = []
  img = that_img
  #get histogram for grayscale value intensity
  hist = np.histogram(img, bins=256)
  #resize image
  img = resize(img, (400,400))
  #extract features but do not yet add them to feature vector
  detector2.detect_and_extract(img)
  #extract HOG features, add them to featurevector
  a = fd = hog(img, orientations=9, pixels_per_cell=(32, 32),
        cells_per_block=(1,1), visualize=False)
  #add histogramm to featurevector
  for h in hist:
     fd = np.append(fd, h)
  #if corresponding boolean in uselist is true add features to featureVector --> Feature selection
happens here
  if(useList[0]):
```

```
        detector.detect(img)
        fd = np.append(fd, [np.array(detector.keypoints).flatten()])
    if(useList[1]):
        fd = np.append(fd, detector2.keypoints)
    if(useList[2]):
        fd = np.append(fd, edgeExtract(img, 100))
    if(useList[3]):
        corners =  corner_peaks(corner_harris(img),min_distance=1)
        fd = np.append(fd, corners)
    #get length of featurevector for later operations
    lengthV.append(len(fd))
    #add featureVector list to dataset that is fed into svm
    DataSet.append(fd)
    max = 17373
    lengthV = []
    DataSet2 = []
    #pad dataset with zeroes so that all featurevectors have the same length --> important for svm
    for d in DataSet:
        d = np.pad(d, (0, max - len(d)), 'constant')
        DataSet2.append(d)
        lengthV.append(len(d))
    DataSet = DataSet2
    return DataSet


def edgeExtract(img, bins):
    retVal = []
    #apply vertical and horizontal sobel filters to get two histogramms, once of vertical and once of
horizontal edges
    #vertical
    fs = filters.sobel_v(img)
    #horizontal
    angs = filters.sobel_h(img)

    #compute histograms
    lhist = np.histogram(fs,bins,normed=True,range=(0,1))
    ahist = np.histogram(angs, bins,normed=True,range=(-180,180))
    #fuse histograms into one list
    retVal.extend(lhist[0].tolist())
```

```
    retVal.extend(ahist[0].tolist())
    return retVal

# test stamp img
demo = random.sample(class_s_test,3)
for name in demo:
  # print(name)
  stamp_img = cv2.imread(test_data_dir+"/"+classes[0]+"/"+name)
  img = cv2.resize(stamp_img,(64,64))
  img = np.reshape(img,[1,64,64,3])
  prediction = (model.predict(img) <0.6).astype("int32")
  cv2_imshow(stamp_img)
  if prediction == 0:
    result="Stamp"
  else:
    result="Not Stamp"
  print("Predict: "+result+"    Actual: Stamp")

# test not stamp img
demo = random.sample(class_ns_test,3)
for name in demo:
  # print(name)
  not_stamp_img = cv2.imread(test_data_dir+"/"+classes[1]+"/"+name)
  img = cv2.resize(not_stamp_img,(64,64))
  img = np.reshape(img,[1,64,64,3])
  prediction = (model.predict(img) < 0.6).astype("int32")
  cv2_imshow(not_stamp_img)
  if prediction == 0:
    result="Stamp"
  else:
    result="Not Stamp"
  print("Predict: "+result+"    Actual: Not Stamp")

# test fake img
demo = random.sample(forg_test,3)
for name in demo:
  # print(name)
  forg_img = cv2.imread(test_data_dir+"/"+classes[0]+"/stamp/"+name)
```

```
  cv2_imshow(forg_img)
  forg_img = cv2.cvtColor(forg_img, cv2.COLOR_BGR2GRAY)
  forg_img_processed = processImg(forg_img)
  result = model_2.predict(forg_img_processed)
  # print(result)
  if result == 0:
    result="Fake"
  else:
    result="Real"
  print("Predict: "+result+"    Actual: Fake")
# test real img
demo = random.sample(org_test,3)
for name in demo:
  # print(name)
  org_img = cv2.imread(test_data_dir+"/"+classes[1]+"/stamp/"+name)
  cv2_imshow(org_img)
  org_img = cv2.cvtColor(org_img, cv2.COLOR_BGR2GRAY)
  org_img_processed = processImg(org_img)
  result = model_2.predict(org_img_processed)
  # print(result)
  if result == 0:
    result="Fake"
  else:
    result="Real"
  print("Predict: "+result+"    Actual: Real")
```