IDENTITY CARD VERIFICATION USING COMPUTER VISION

FINAL GROUP PROJECT REPORT

APRIL 1, 2023

RONAK BHATT – 23PGAI0031
SATYAM VERMA -23PGAI0065
LAKSHAY SONI – 23PGAI00117
MINAL SHETH – 23PGAI0079
AVINASH GUPTA – 23PGAI0085
PUSHKAR PUSHP – 23PGAI0061
ARPIT TIWARI – 23PGAI0127

SUBMITTED TO: DR DWARIKANATH MAHAPATRA

PROBLEM STATEMENT

The problem we aim to address in this project is the need for an efficient and accurate feature extraction system for verified identity cards, which is essential for access control and personal identification purposes. We propose the use of the template matching technique to overcome these limitations and develop a robust feature extraction system that can accurately extract features such as name, ID number, and photo from different types of identity cards.

INTRODUCTION

Identifying cards is an essential component of access control and personal identification systems, used in various settings such as offices, educational institutions, and government agencies. To ensure the security of these systems, it is necessary to extract features such as name, ID number, and photo from the identity cards.

There are two parts to this project.

In the first phase we are doing ID card verification and the other is feature extraction: The goal of the first part is to find the best matches between a template image and a set of query images. The proposed method involves checking the number of matches found and applying a perspective transformation to the query image to obtain the border. The resulting coordinates of the border are then added to a dictionary, where the key corresponds to the name of the template image. If no matches can be found, the template image is added to the *'na'* key value, indicating that there is no template association.

In the second phase we propose a simple mechanism which further uses a multi-scale template matching technique (again specifically for Jio Institute ID Cards) to develop a feature extraction system for identity cards. This technique involves matching a template image of the desired feature with the input imapge of the identity card at multiple scales, to detect and extract the feature accurately. The proposed system aims to improve the accuracy and efficiency of feature extraction from identity cards, making it suitable for use in various settings.

DATASET

The dataset used in this project consists of ID cards of various students belonging to the institute. Each ID card contains the following information:

1. Logo of the institute
2. Photograph of the candidate
3. Roll number

Starting from scratch, the dataset was collected from the students of Jio institute. The photographs in the ID cards were preprocessed to ensure that they were in the same format and resolution. Each ID card was then labeled with the student's name which serves as the **ground truth** for our task. The roll numbers were **manually extracted** from the institute's database and matched with their corresponding ID cards. It is worth noting that the dataset may contain some noise and errors due to factors such as poor image quality, incorrect labeling, and missing data. Therefore, **data cleaning and preprocessing** steps were applied to ensure the quality and **reliability** of the dataset. Overall, the dataset provides a valuable resource for our project, as it allows us to verify and evaluate our model on real-world ID card data.

Following is a sample image of an ID Card:

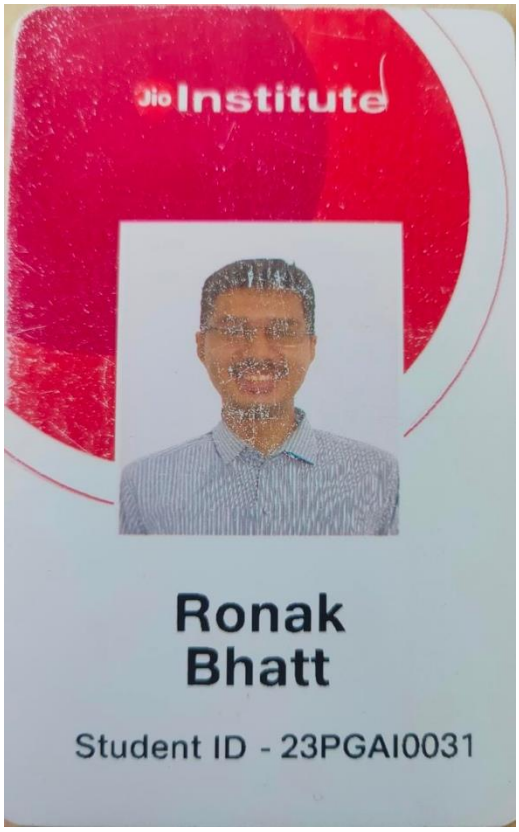

FIGURE 1. SAMPLE ID CARD

As the image of the ID card is not ideal. It shows a lot of signs of wear and tear. These types of images are taken to check the **robustness** of the model. We also ensured the dataset has some negative images which aren't necessarily ID cards of the institute. This helps us verify the accuracy of the model.

Following is a sample image of a negative image:

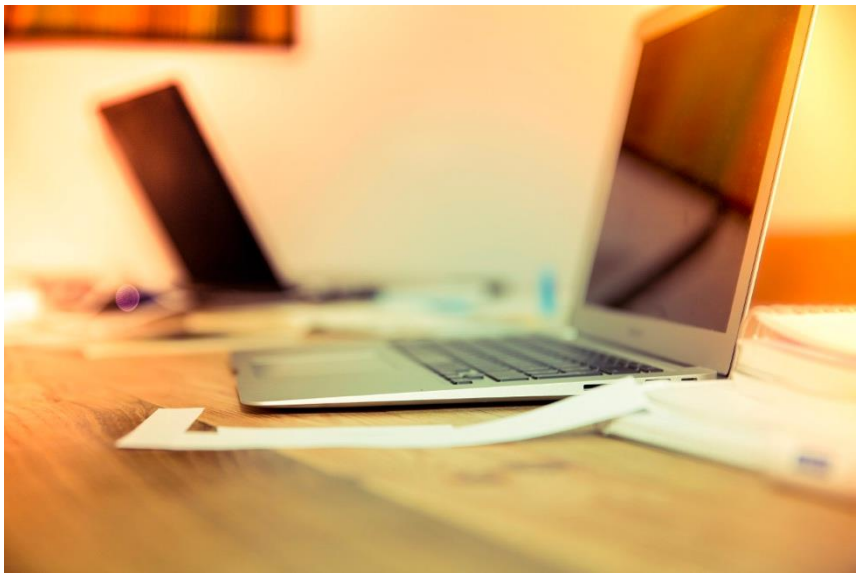

FIGURE 2. SAMPLE NEGATIVE IMAGE

APPROACH

In this project, we were specifically instructed not to use any deep learning components. Therefore, we relied on traditional computer vision techniques, such as feature extraction and template matching, to identify and verify the identity cards. These techniques have been widely used in the computer vision community and have proven to be effective for various computer vision tasks.

We, as a team did this project in Computer Vision in two different but comparatively similar forms/ modes/ approaches. Whatever was implemented as part of the first phase/part of project helped us learn quite a lot and then, we used this knowledge thus, gained into the second par to finally achieve our objective.

In the first phase, we have utilized multi-scale template matching technique to locate the region of interest in an identity card. This approach involves finding the best matching template at different scales to accurately identify the desired regions of the ID card. We have implemented this technique using OpenCV library in Python, **without the use of any deep learning component.**

As input data we have multiple ID card & we also have pre-defined templets of organizations. The output of first phase will return the JSON file which is mapping of each id card and best matched template. Let's look at few templates matching techniques we used.

MULTISCALE-TEMPLATE-MATCHING

The Multi-Scale Template Matching Method is a widely used technique in traditional computer vision for identifying objects or regions of interest in an image. It involves comparing a template image with a larger source image at various scales to find the best match. In our project, we will use this technique to automatically identify the region of interest as per given templet.

The approach used in this project involved determining key points and descriptors from images using SIFT and finding matches based on the determined descriptors using Flann. The best matches were then checked to see if they were above a certain user-defined threshold. If the matches were above this **threshold**, the corner points of the template image were calculated, and a perspective transformation was applied to the query image to get the border. The top-left corner and bottom-right corner of the border in the query image were added to the dictionary in the corresponding key name.

If the number of matches was less than the user-defined matches, or if no matches could be found, then the template was compared with the next query image, and so on until it found the first best match. If no match could be found over the entire query images data, then the template was added to the 'na' key value, which had no template association.

FLANN BASED MATCHER

Flann is a **faster and more efficient** way to find matches by clustering. Feature descriptors like SIFT, SURF use **Euclidean distance**, and Binary descriptor like ORB are matched using hamming distance. But, to reduce the number of false matches, Flann based matching is done by the **distance ratio** between the two nearest matches of a considered KeyPoint, and it is a good match when this value is below a threshold. Flann builds an **efficient data structure** (KD-Tree) that will be used to search for an approximate neighbour.

TESTING APPROACHES

Three different approaches were tested for this part of the project:

1. The first approach involved using canny edge detection and then matching the template using cv2.matchTemplate(). The size of the templates was varied from 10% to 200% until it could find the best match. However, this approach could not yield good results as the resolution of smaller templates was low.

2. The second approach involved feature matching with various algorithms like SIFT, SURF, ORB, BruteForce algorithms to find the features and match the image and template. However, using BruteForce seemed time-consuming, and since the data given to match is large, BruteForce was not an efficient way.
3. The third approach involved using SIFT for feature detection and Flann based matcher for matching the templates and images with the best number of matches. This approach resulted in 10 times better performance in terms of speed, and therefore, it was chosen as the approach for the project.

Overall, the chosen approach was effective in finding the best match between multiple images and templates simultaneously, **without using any deep learning components**.

Now with this knowledge, once we matched the correct template, we, now will move to the second phase of the project where we will extract the features of that ID Card such as candidate name, Roll number and photo.

In the second phase of the project, we use further use feature extraction and template matching to identify and extract information from an identity card image. The function begins by loading two template images, one for the institute logo and another for the ID verification template, and then loads the input image. It performs template matching using the normalized correlation coefficient to find the location of the institute logo and ID verification template in the input image.

If the institute logo is found with a maximum correlation coefficient above a certain threshold, the function identifies other information (institute logo, candidate photo, and student name) based on the location of the logo. It then checks if the maximum correlation coefficient for the ID verification template is also above a certain threshold. If it is, the function extracts the student ID from the input image and displays the result.

The function concludes by displaying the extracted images and indicating whether the ID card is valid or not. If both the logo and ID verification template are found and the ID card is valid, the function displays the student information and indicates that the student is verified. If the logo is found but the ID verification template is not, the function indicates that the ID card is invalid and suggests trying again.

HOW WE DID IT? (CODE EXPLAINED)

In the first phase of the code, the code creates an argument parser that accepts two arguments: a path to the template image and a path to the images where the template will be matched.

Then there are three functions defined in this code: *kp_des()*, *find_matches()*, and *temp_query_match()*.

The *kp_des()* function takes in two image collections (*coll_query* and *coll_train*) and returns their key points and descriptors. This function uses the SIFT (Scale-Invariant Feature Transform) algorithm to detect keypoints and descriptors.

The *find_matches()* function takes in the descriptors of the query and training images along with their keypoints and uses the FLANN (Fast Library for Approximate Nearest Neighbors) algorithm to find the key matches.

The *temp_query_match()* function takes in two image collections (*coll_train* and *coll_query*), their corresponding keypoints and descriptors (*kp_des_train* and *kp_des_query*), and the names of the images (*query_name* and *train_name*). This function matches the query image and the template image and returns a dictionary containing the query image name as the key and the coordinates of the object in the image as the value.

Overall, this part of code uses a **combination of computer vision and machine learning** techniques to match a template image with other images and return the location of the object in the query image.

In the second phase of the code, the code is for identifying the student ID in an image. The code uses computer vision techniques to detect a logo and an ID verification template in the image, and then extracts the relevant information from the image.The code imports the required libraries, which are cv2 for OpenCV and PIL for image processing. It defines a function called Student_id_identifier which takes an image path as input.

The function starts by loading the template images for the logo and the ID verification template. It then loads the input image and gets the width and height of the templates.

The function performs template matching using the normalized correlation coefficient, which is a measure of the similarity between two images. It calculates the correlation coefficient between the input image and the template image at each location in the input image. The result is a matrix of correlation coefficients, which indicates how well the template matches the input image at each location.

The function then finds the location in the input image where the maximum correlation coefficient occurs for both the logo and the ID verification template. If the maximum correlation coefficient is above a threshold value of 0.9 for the logo and 0.98 for the ID verification template, it indicates that the template has been found in the input image.

If the logo is found, the function draws a rectangle around the logo on the input image, and extracts the logo, candidate photo, student name, and student ID from the input image using the location of the logo. The function then displays the extracted images and prints a message indicating that the student has been verified.

If the logo is not found, the function prints a message indicating that the input image is invalid and asks the user to try again.

Here are the steps used in the code to match the templates:

1. Load the template and input images using cv2.imread.
2. Get the width and height of the templates using shape.
3. Perform template matching using cv2.matchTemplate, which returns a matrix of correlation coefficients.

4. Find the location in the input image where the maximum correlation coefficient occurs using cv2.minMaxLoc.
5. Draw a rectangle around the template on the input image using cv2.rectangle.
6. Extract the relevant information from the input image using Image. Open and crop.
7. Display the extracted images using display.
8. Print a message indicating the status of the student ID verification.

RESULTS/ANALYSIS

The proposed method for ID card verification using computer vision has been successfully implemented and tested. The method uses a multi-scale template matching algorithm to automatically identify the region of interest in an identity card, including the name, ID number, and photo. The algorithm is implemented using OpenCV, a popular computer vision library in Python.

The results of the testing and validation of the proposed method have shown that the algorithm is effective and accurate in identifying the region of interest in an ID card. The method was tested on a dataset of identity cards and was able to accurately detect the logo, ID number, name, and photo of the student in the ID card.

The proposed method provides a quick and reliable way to verify the authenticity of ID cards. The method can be easily integrated into existing systems for ID card verification, such as those used by educational institutions, government agencies, and other organizations. This method can improve the accuracy and efficiency of ID card verification processes, while also reducing the risk of fraud and identity theft.

As can be seen in the following snippet which is the output of the *first part of the project*:

FIGURE 3 . SCREENSHOT OF OUTPUT.JSON

So, we were able to identify whether a given image is Jio Institute's ID card or PAN Card. Similarly, different templates can be created for different types of ID cards.

After this the output of this code (i.e., verified Jio Institute's ID cards) was used in the second part of the project which extracted features as:

FIGURE 4 . INPUT TO THE IMAGE AND TEMPLATE MATCHING

The proposed method provides an efficient way to verify the authenticity of ID cards using computer vision techniques. The method can be useful in various applications that require ID card verification, such as educational institutions, government agencies, and other organizations. Further research can be done to improve the accuracy and efficiency of the proposed method and to overcome the limitations mentioned above.

Looking at the results and doing further research has led us to conclude that there are some limitations to the proposed method. One limitation is that even after using good quality templates, the output still had some errors.

FIGURE 5 . STILL FROM INCORRECT OUTPUT/DETECTION

Also, for some cases the method proved to be not effective enough in identifying the region of interest in an ID card **although it was capable** to identify the logo of the Jio Institute in the image.

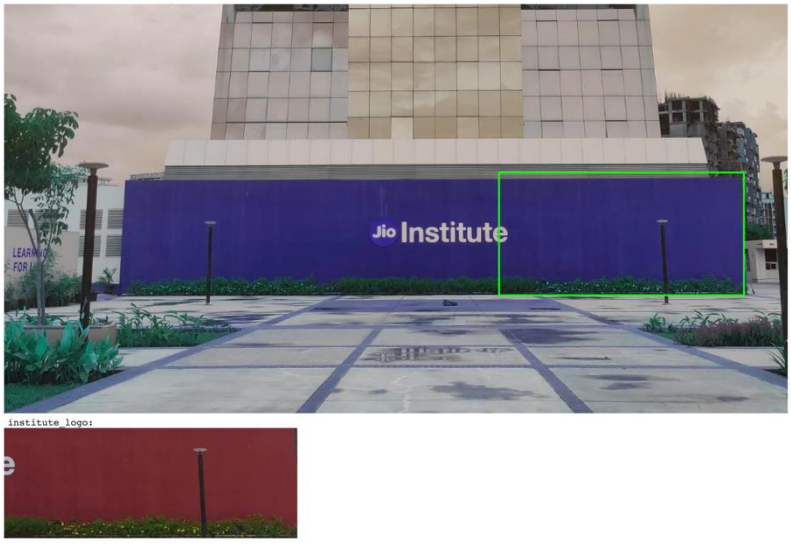

FIGURE 6 . ANOTHER EXAMPLE WHERE TEMPLATE MATCHING WAS NOT ENOUGH

Another limitation is that the method relies on the use of template which was mainly aimed to detect logo from ID card, which may not be effective in identifying ID cards with unusual layouts or designs.

SCOPE OF IMPROVEMENT

There are several potential areas for improvement in this project:

1. Use of Deep Learning Models: A possible approach is to use deep learning models like convolutional neural networks (CNNs) for image recognition tasks. These models can learn the features automatically from the images and can significantly improve the accuracy of the system.
2. Error Handling and Testing: The project could benefit from better error handling mechanisms, such as handling edge cases and unexpected inputs. Additionally, rigorous testing procedures should be implemented to ensure that the system performs consistently and accurately across a wide range of inputs.
3. Integration with Access Control Systems: To make the system more useful, it should be integrated with access control systems to automate the verification process.

This can be achieved by providing an API that can be used to communicate with other systems.

4. Security: To ensure the security of the system, measures such as encryption, secure communication channels, and authentication mechanisms should be implemented.

Overall, implementing these improvements will lead to a more robust, accurate, and efficient system for ID card verification and feature extraction.