Analysis of Flip/Mirror slide labels
{Label classification using Image Morphing Techniques}

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology
in

Computer Science and Engineering

By
Aditya Saxena [191269]

Under The Supervision of

Dr. Ruchi Verma

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

Certificate

I hereby declare that the work presented in this report entitled **"Analysis of Flip/Mirror slide labels {Label classification using Image Morphing Techniques}"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2023 to May 2023 under the supervision of **Dr. Ruchi Verma** (Assistant Professor (SG), Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat).

I also authenticate that I have carried out the above mentioned project work under the proficiency stream **Data Science**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Aditya Saxena

Aditya Saxena [191269]

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Ruchi 10/May/23

Dr. Ruchi Verma
Assistant Professor (SG)
Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology
10 May 2023

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 15/05/23

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report ✓ | Paper

Name: ADITYA SAXENA Department: CSE Enrolment No 191269

Contact No. 7599003687 E-mail. saxenaaditya12@gmail.com

Name of the Supervisor: Dr. RUCHI VERMA

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ANALYSIS OF FLIP/ MIRROR SLIDE LABELS {LABEL CLASSIFICATION USING IMAGE MORPHING TECHNIQUES}

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 44
- Total No. of Preliminary pages = 9
- Total No. of pages accommodate bibliography/references = 1

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 5.0 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
			Word Counts	
Report Generated on				
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

Acknowledgement

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this project.

I am grateful to my faculty supervisor, **Dr. Ruchi Verma, Assistant Professor (SG), Dept. of CSE & IT, Jaypee University of Information Technology, Waknaghat,** for her invaluable guidance, support, and encouragement throughout the project.

I also extend my sincere gratitude to my company supervisor, **Ms. Priyanka Golchha, Staff Engineer, nference**, for her consistent support at each step of the project. Her expertise in the field of Data Science has been instrumental in shaping the project and bringing it to fruition. Her constructive feedback and insightful suggestions have been instrumental in refining the project to its present form.

I would also like to extend my gratitude to my colleagues at nference and staff members of JUIT, who have provided me with their timely support, assistance, and resources, which have been crucial to the success of this project.

I am also thankful to the almighty for granting me the strength and perseverance to complete this project. Finally, I would like to acknowledge with deep respect and gratitude the constant support and encouragement of my parents. Their unwavering support, patience, and love have been a source of strength and inspiration throughout the project.

Aditya Saxena

Aditya Saxena
191269

Table of Content

List of Abbreviations

Abbreviations	Full form
DICOM	Digital Imaging and Communications in Medicine
UI	User Interface
OCR	Optical character recognition
NER	Named entity recognition
DID	De-Identification

List of Figures

List of Diagrams

Abstract

Pramana, an nference initiative, revolutionizes whole slide imaging (WSI) by using proprietary software and scanning hardware to generate highly accurate images. Pramana's cost-effective and turnkey DPaaS archival services unlock the power of historical anatomical pathology, providing researchers and scientists with millions of glass slides that were previously limited in utility. This breakthrough technology meets the future challenges of pathologist shortages and increasing volumes of tissue biopsies, enabling drug and algorithm development. In addition to pioneering a new whole slide imaging paradigm, Pramana's DPaaS archival services have the potential to significantly impact research and innovation in the field of anatomical pathology. These pathology glass slides have rich metadata associated with them in the form of labels. The labels contain printed text, handwritten text, barcodes and several kinds of annotations. Thus, Image processing, NER, OCR and various other techniques are utilized in obtaining this data.

This project discusses one major challenge faced during the archival process, which is to check whether the slide kept in the scanner is flipped (or has mirror content on it). It is a major challenge as the time taken for successful archival is a crucial factor affecting the company’s revenue generation, and if the slide is flipped, correct data can not be gathered and hence resulting in unsuccessful archival plus wastage of time for scanning an incorrect slide.

The algorithm involves detecting barcodes on the label, followed by checking whether the label has mirror content on it or not, and then finally detecting whether the label is blank or has some impression over it. All the labels are passed through this pipeline and ultimately the class of the label is detected on the basis of certain thresholds at each step.

Chapter 01: Introduction

1.1 Introduction

'nference' is an AI-driven healthcare company that leverages advanced machine learning and natural language processing algorithms to extract meaningful insights from complex biomedical data. With a mission to accelerate drug discovery and improve patient outcomes, nference offers a suite of solutions that include data integration and harmonization, target discovery, drug repurposing, clinical trial optimization, and more. Their proprietary platform is trusted by leading pharmaceutical companies, academic institutions, and healthcare organizations worldwide to uncover new therapeutic opportunities, streamline research workflows, and advance precision medicine.

Pramana is a pioneering initiative by nference that provides innovative solutions to the field of anatomical pathology. Pramana has redefined whole slide imaging (WSI) with their innovative technology designed to streamline lab operations, empower pathologists, and facilitate algorithm development. Using pixel level, dynamic z-stacking, automated, in-line quality control (including rescanning and AI-driven tissue detection), Pramana's proprietary software and scanning hardware generates highly accurate images, boasting a success rate of greater than 99.5%. This exceptional image quality instills confidence in pathologists to adopt Digital Pathology. Furthermore, Pramana's cost-effective and turnkey DPaaS archival services unlock the potential of millions of glass slides that were previously limited in utility, facilitating research and innovation. By addressing future challenges such as pathologist shortages and increasing volumes of tissue biopsies, Pramana enables drug and algorithm development, thus revolutionizing the field of anatomical pathology.

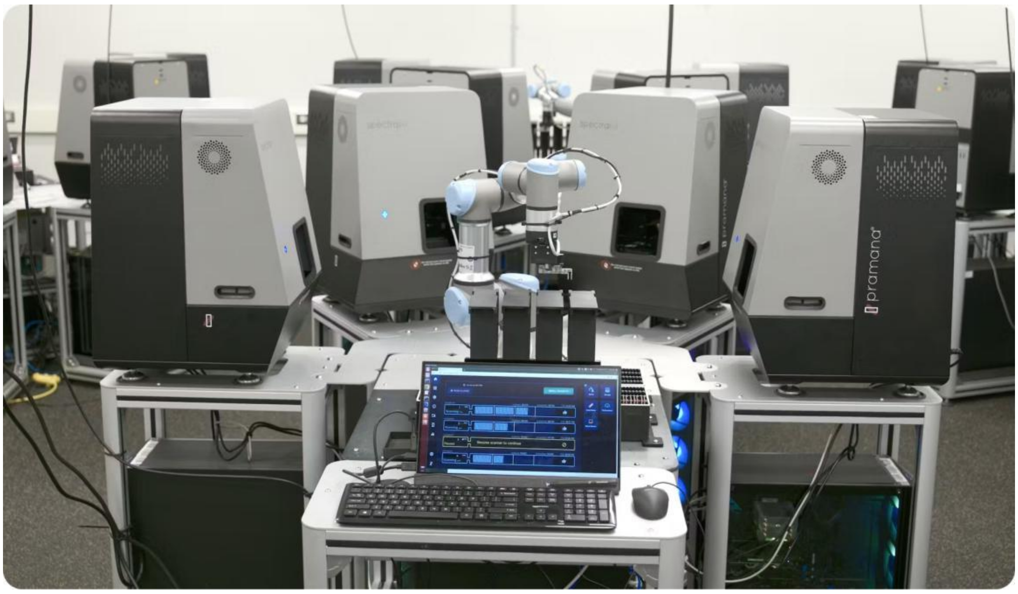


Fig. 1: Image of a Pramana cluster.

Description of a Pramana cluster:

The design of a Pramana cluster is composed of four scanners, each connected to a high-end CPU, one robot with its own CPU, cluster storage, one master system for managing the entire cluster, slide baskets, drop baskets, and cooling systems. The robot picks up a glass slide from the basket and feeds it into the scanner. The scanner performs whole slide image (WSI) analytics, and once a slide is scanned, it is shifted to the drop basket by the robot. The robot continues to perform the same operations with the remaining slides.

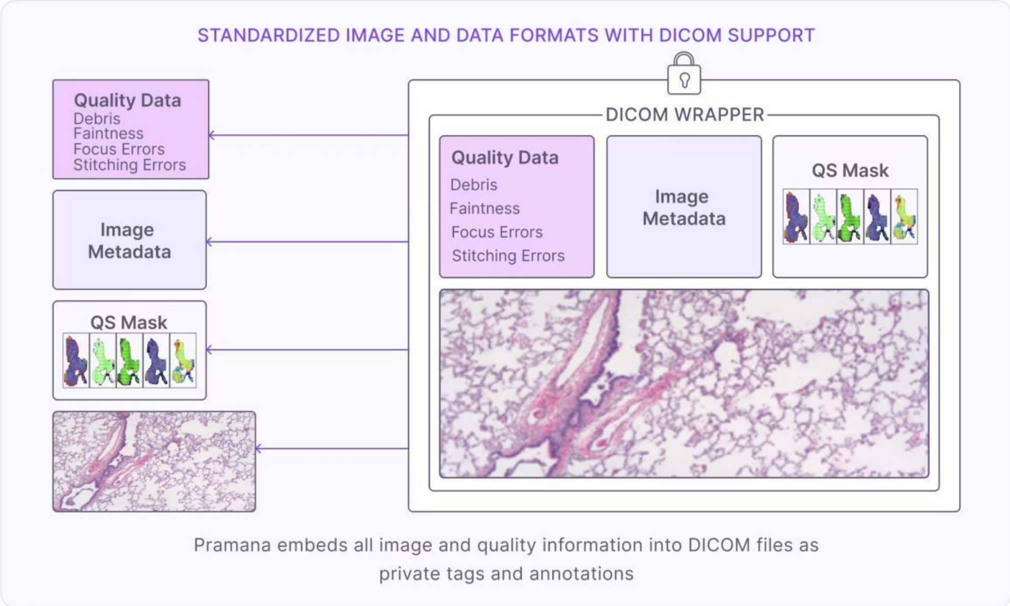


Fig. 2: Types of data extracted from a glass slide.

Description of a glass slide:

A pathology glass slide consists of two regions. The top portion of the slide has a label that contains important information about the slide, such as tissue, stain type, person's details, etc. This section is referred to as the metadata of the slide. The remaining portion consists of the tissue sandwiched between glass.

For this project, the focus is on the top portion of the glass slide, which consists of the label with metadata of the slide.

Fig. 3: Sample label image as captured by the scanner camera.

Description of the label:

The label is one of the most important parts of a glass slide, without any metadata the archival process is incomplete. The metadata helps in identifying the tissue type, the process with which it was stained and a lot of other important details which are crucial for the other algorithms to function. This metadata also helps in developing and training predictive algorithms to make the life of doctors and pathologists much easier and to enable highly accurate diagnosis.

The label contains Case ID, Block ID, Slide ID, Name of the patient, tissue type, stain type, printed text, handwritten text and annotations, Clinic and location where the sample was taken, date, and barcodes with more metadata. To extract this information the 1x image of the label is processed and then applied with OCR, NER and other morphological operations to detect and decode barcodes.

The company follows a strict rule to not share medical information of an individual in any way, and thus there is a step of De-Identification (DID) of a slide before it is used for training or research purposes. This step also relies on correct detection and localisation of label data, as NER is applied to detect personal information on the label and then de-identifying the same.

For 100 years, the process of making a glass slide has been almost the same, and after a certain time period the tissue decays and can not be analyzed and thus digitalising such slides becomes important for the advancement in the field of medicine and healthcare. The glass slides consist of a variety of combinations and arrangements of the label metadata elements mentioned above, and to tackle all these variations smart algorithms have been developed so that none of the case is missed and all the abundant historical data can be archived.

These slides are continuously being scanned using the Pramana scanners around the world, and time to scan each slide effectively is a crucial factor deciding the revenue generation of the company, the efficiency of the archival process and the overall performance of the Pramana technology. As human error is an inevitable factor in each field, smart solutions are being developed to tackle each of them using technology. In this project one such human error, of keeping a glass slide upside down is being handled, which in return will prevent wastage of time for incorrect scanning and alert the user for correcting the orientation of all the slides.

1.2 Problem Statement

While scanning the pathology glass slides, sometimes the slides are placed flipped in the slide basket, or at times the entire basket is kept in the opposite orientation than expected. This issue wastes time spent for scanning the slides as incorrectly placed slides do not result in successful archival. With this issue lots of services are compromised due to the incorrect data being fed from the scanner end. This problem directly affects the revenue collection of the company as well.

The label that is placed may have printed text, handwritten text, barcodes, combination of all of these, or at times it can be blank as well. The handwritten text present on the slide can not be detected with high confidence by OCR modules. The printed text contains symmetrical alphabets at times and even when the slide is kept flipped, the OCR can make sense out of such words. The problem becomes even more complex as sometimes there are some shapes present on the labels marking the correct side, which can not be detected by any OCR or barcode utility.

1.3 Objective

With such a variety of cases being present, there is a need to construct a pipeline in order to manage each of them step by step and classify them under the correct classes. The objective includes to use minimum time to detect the slide orientation and further save even more time by passing only the correct slides for scanning.
Ultimately the main objective is to pass a slide to scan if it is placed correctly and prevent an incorrectly placed slide to be sent for scanning.

1.4 Motivation

The accurate scanning and digitization of pathology slides are basic for proficient and compelling medical diagnosis and treatment planning. However, error in slide orientation during scanning can prompt erroneous information being documented, which can compromise the quality of patient care. Such errors can result in wasted time and resources, as well as lost income for the organization. Current solutions such as manual checking are time-consuming and prone to human error. In addition, the presence of manually written text and even printed text on slide marks can present huge difficulties for OCR modules. To resolve these issues, there is a pressing need to develop an automated system that can detect slide orientation, classify slides based on their content, and streamline the scanning process. The proposed framework plans to limit the time expected to recognize slide orientation, and ensures that only correctly oriented slides are passed for scanning. Ultimately, this system has the potential to enhance the accuracy and efficiency of medical diagnosis and treatment planning, while reducing costs and improving patient outcomes.

1.5 Methodology

The slide is placed in the slot and a 1x image of the label is captured. This image is taken as input for our pipeline and analyzed for its orientation. OCR is applied to the image and a text list with confidence value of each is generated. Barcode detection module localizes and decodes if any barcode is found on the image. If the decoded barcode value is not empty, then it is confirmed that the image is not flipped and thus it is marked as a non flipped slide and is sent for scanning. If there is no barcode value decoded then we analyze if the label is a mirror image.

For analyzing if the label is a mirror image or not, we apply a flip operation on the image and make a hypothesis that the original image was a normal image, and after the flip operation the image is mirrored. The mirror image is sent through an image processing pipeline consisting of operations such as blurring, normal thresholding, adaptive thresholding, and denoising. After these operations, OCR is applied on this version. Through a series of testing Paddle OCR was found to produce promising results and thus the same is used in the final pipeline. After this step we receive a list of words found on the label along with their confidence values. We pass these lists of words and confidence values, and the same from the initial step of applying OCR on the original image, to our decision block where we compute total number of text boxes, total number of above threshold boxes, length of high confidence words (for cases with only 1 or 2 text boxes on the label). The decision block makes decision on the following conditions:

1) If no or few boxes are found on the label, we mark it is an ambiguous case.
2) If both versions have an equal number of high confidence boxes, we mark those labels as ambiguous as well.
3) If the original image has more high confidence words, our hypothesis stands true, and we mark the image as not flipped, and further send it for scanning.
4) If the mirror image has more high confidence words, our hypothesis fails, and we mark the image as mirrored and it is prevented from scanning.

Most of the cases are handled with this step of the pipeline, but at times certain cases have no readable text or no text at all, and so all the ambiguous cases are moved to the next step of the pipeline.

Next we analyze whether the image is empty or not based on the pixel intensity of the image. For this step certain image processing steps are applied before coming to a decision. As the label image has frame backgrounds on the corners and dark shadows of the machinery parts on the edges, we first crop the image to have focus on the label region. To remove the dark shadows of the machinery we apply a blackhat operation on the image, which gives an image with enhanced dark regions of interest on a light background, and handles brightness gradient over the label image produced due to direction of light while capturing. Now the image is suitable for applying thresholding. Two types of thresholding are applied on the label image, normal thresholding and adaptive thresholding. The process of simple thresholding involves assigning a global threshold value to all the pixels in an image. However, adaptive thresholding is a technique where smaller regions are evaluated to determine the threshold value. As a result, varying threshold values are assigned to different regions within the image.The use of adaptive thresholding in this case is to enhance and focus on the lighter pixel regions clearly, which might be missed due to normal thresholding. Again, adaptive thresholding focuses on small dust or grain on the images as well and results in a lot of noise on the image, which is then handled by applying a morphological closing operation. The closing operation is a common morphological operation in image processing that involves two sequential steps: dilation and erosion. Dilation is the process of expanding the boundaries of an object in an image by adding pixels to its perimeter. Erosion, on the other hand, is the process of shrinking the boundaries of an object by removing pixels from its perimeter. The closing operation uses both of these operations in succession, with the dilation step followed by the erosion step, using the same structuring element for both operations. The black pixel intensities are computed on both normal thresholded image and adaptive thresholded image. These black pixel intensities are compared with a threshold value, which was computed by running several experiments over a variety of labels. If the black pixel intensities are higher than threshold, then we say that the label is not blank, and we mark it as normal and pass it for scanning. If the black pixel intensities are lower than threshold, then we say that the label is blank or flipped, and we prevent it from sending it for scanning.

Chapter 02: Feasibility Study

2.1 Feasibility Study

Literature Survey

Mirror image detection is a significant undertaking in image processing and computer vision applications. This issue has been concentrated on in different spaces, including archive examination, biometrics, and article acknowledgment. In the space of record examination, mirror image detection is a pivotal undertaking to guarantee the exactness of optical character recognition (OCR) frameworks. In biometrics, mirror image detection is utilized to forestall extortion in facial recognition frameworks. In object recognition, mirror image detection can be utilized to separate among left and right-handed objects.

Different methods have been proposed in the writing for mirror image detection. A portion of the strategies depend on feature extraction and classification, while others depend on template matching and connection. Lately, profound learning-based methods have additionally been proposed for mirror image detection. One of the well known strategies for mirror image detection depends on feature extraction and characterization. In this methodology, highlights are removed from the picture and afterward grouped utilizing a classifier, for example, support vector machine (SVM), k-closest neighbor (KNN), or choice trees. One more methodology depends on format coordinating and relationship. In this methodology, a layout picture of the item is made and afterward coordinated with the information picture to decide the comparability. As of late, profound learning-based procedures have shown promising outcomes for mirror image detection.

[1] "A Study on Detection of Flipped Images in Text Data," by Seung-Ki Ryu et al. (2018). This paper presents a calculation for recognizing flipped pictures in message information by dissecting the bearing of message lines.

The calculation utilizes a mix of edge identification, morphological tasks, and Hough change to identify text lines and their headings. The creators guarantee that their calculation accomplishes high exactness in identifying flipped pictures.

[2] "Automatic Detection of Mirror Images in Medical Images," by Sushma Guduru et al. (2016). This paper presents a calculation for distinguishing mirror images in clinical pictures utilizing edge discovery and picture likeness measures. The calculation analyzes the edges of the original and mirror image and figures their comparability scores. The creators guarantee that their calculation can identify perfect representations with high exactness.

[3] "Mirror Image Detection Using Convolutional Neural Networks," by Ryohei Hisano et al. (2018). This paper presents a profound learning-based approach for distinguishing identical representations utilizing convolutional neural networks (CNNs). The creators utilize a pre-prepared CNN model and tweak it on a dataset of mirror and non-identical representations. They report high precision in recognizing identical representations on their dataset.

[4] "Detection of Mirrored Text in Images," by Tarun Pruthi et al. (2018). This paper presents a strategy for identifying mirrored text in pictures utilizing message bearing examination and stroke width variety investigation. The creators initially distinguish text locales in the picture and afterward break down the course and stroke width varieties of the text to decide whether it is reflected or not. They guarantee that their technique accomplishes high exactness in recognizing reflected text.

[5] "A Novel Approach to Mirror Image Detection in Printed Documents," by Mohammad Muzahidul Islam et al. (2020). This paper presents a

calculation for recognizing mirror images in printed records utilizing a blend of edge location, morphological tasks, and line fragment recognition. The creators guarantee that their calculation can distinguish perfect representations in printed records with high exactness.

In outline, mirror image detection is a significant undertaking in picture handling and PC vision applications. Different methods have been proposed in the writing, including highlight extraction and characterization, format coordinating and connection, and profound learning-based strategies. The decision of method relies upon the particular necessities of the application, like precision, speed, and robustness.

Chapter 03: System Development

3.1 Algorithm Overview

The process begins by taking a picture of a slide label in a slot and determining its orientation. Using OCR, a list of text is generated with confidence values. The algorithm then detects and decodes any barcodes present on the label. If a barcode is decoded, the label is considered non-flipped and sent for scanning. If not, the algorithm checks if the label is a mirror image by flipping the image and processing it. OCR is applied to the second version as well, and the algorithm uses the results to determine if the label is ambiguous, not flipped, or mirrored. If the label is ambiguous, the algorithm analyzes the pixel intensity of the image to determine if the label is empty or not. The label region is enhanced using various image processing operations, and pixel intensities are compared to a threshold value to determine if the label is normal or blank/flipped.

OCR

PaddleOCR support various state of the art algorithms connected with OCR, and created industrial featured models/solution PP-OCR on this premise, and overcome the entire course of information creation, model training, compression, derivation and deployment.
PP-OCR is a self-created useful super lightweight OCR framework, which is smeared and streamlined in view of the reimplemented scholastic calculations, taking into account the harmony among precision and speed.

PP-OCR is a two-stage OCR framework, wherein the text location calculation is DB, and the text recognition algorithm is CRNN. Plus, a text course classifier is added between the localisation and recognition modules to manage text in various directions.

PP-OCR pipeline is as per the following:
PP-OCR framework is in constant streamlining. As of now, PP-OCR and PP-OCRv2 have been delivered: PP-OCR embraces 19 powerful procedures from 8 angles including spine network determination and change, forecast head plan, information expansion, learning rate change technique, regularization boundary choice, pre-preparing model use, and programmed model fitting and quantization to enhance and thin down the models of every module (as displayed in the green box above). The eventual outcomes are a super lightweight Chinese and English OCR model with a general size of 3.5M and a 2.8M English computerized OCR model.

NER

Named Substance Acknowledgment (NER) is a well known information preprocessing task in Normal Language Handling (NLP) that includes distinguishing key data in text and ordering it into predefined classes. An element is regularly something reliably discussed or alluded to in the text.

At its center, NLP is a two-step process that includes identifying elements from text and characterizing them into various classifications. The absolute most significant classes in NER are individual, association, and spot/area. Other normal errands incorporate ordering date/time, articulations, numeral estimations (e.g., cash, percent, weight), and email addresses.

Techniques for NER incorporate preparation models for multi-class arrangement utilizing different AI calculations. Be that as it may, this approach requires a ton of naming and a profound comprehension of setting to manage sentence equivocalness, making it trying for a basic AI calculation. Another technique is Contingent Irregular Field (CRF), a probabilistic model that models successive information like words and can catch a profound comprehension of sentence setting. Profound Learning Based NER is a more exact strategy that utilizes a procedure called word implanting to figure out the semantic and syntactic connections between different words. It can likewise investigate theme explicit and undeniable level words naturally, making it relevant for different undertakings.

Furthermore, profound learning can do the vast majority of the tedious work itself, permitting scientists to effectively utilize their time more.

Image Processing

Digital image processing is the utilization of computer calculations and numerical models to process and investigate computerized pictures. This innovation intends to upgrade the nature of pictures, separate significant data from them, and automate picture based tasks. The interaction includes a few central stages, including image acquisition, enhancement, restoration, segmentation, representation and description, analysis, and synthesis and compression. Digital image processing is utilized in different applications, like clinical imaging, remote detecting, computer vision, and mixed media.

Digital pictures can be arranged into various kinds in view of their pixel components and variety designs. Binary pictures contain just two pixel components, 0 and 1, addressing highly contrasting, individually. High contrast pictures comprise of just high contrast tones. The 8-bit color format, otherwise called grayscale, has 256 unique shades of varieties. The 16-digit variety design, otherwise called high tone, has 65,536 distinct varieties. The RGB design partitions the 16-cycle design into three further arrangements: red, green, and blue.

In summary, digital image processing is a strong innovation that considers the improvement and automation of picture based tasks. By understanding the various sorts of digital pictures and their configurations, specialists can all the more precisely process and dissect advanced pictures for different applications.

Morphological Image Processing

Numerical Morphology is a strong strategy used to remove picture parts that are valuable in addressing district shape, limits, and that's only the tip of the iceberg. This method depends on an extensive arrangement of picture handling tasks that cycle pictures in light of shapes. In a morphological activity, an organizing component is applied to an information picture, making a result picture of a similar size. The worth of every pixel in the result picture is resolved in light of a correlation of the relating pixel in the info picture with its neighbors.

Morphology and Picture Division are firmly related, with morphology frequently used to pre-process input information for Picture Division or to post-process the result of the Picture Division stage. Morphological tasks can eliminate defects in the fragmented picture and give data on the shape and construction of the picture.

The terms utilized in morphological picture handling depend on the organizing component. The organizing component is a lattice or a little measured format used to navigate a picture. At the point when every one of the pixels in the organizing component cover the pixels of the article, we consider it a "Fit." When something like one of the pixels in the organizing component covers the pixels of the item, we consider it a "Hit." When no pixel in the organizing component covers the pixels of the item, we consider it a "Miss."

Morphological picture handling is like spatial sifting, with the organizing component got across each pixel in the first picture to make another handled picture. The two most generally utilized tasks are Disintegration and Enlargement. Disintegration shrivels the picture pixels, while enlargement grows them. The result pixel values are determined in light of whether the organizing component fits or raises a ruckus around town. These activities can be utilized to remove significant data from pictures and are generally utilized in different applications, including clinical imaging, remote detecting, and computer vision.

Top Hat and Black Hat Transform

In morphology and advanced picture handling, the top hat and black hat transforms are significant tasks for removing little components and subtleties from pictures. The top hat transform is accomplished by taking the contrast between the info picture and its initial utilizing an organizing component, while the black hat change is the distinction between the end and the info picture. These changes are utilized for different picture handling undertakings, like component extraction, foundation adjustment, and image upgrade.

The top hat channel is especially helpful for upgrading bright objects of interest against a dull foundation. Then again, the black hat activity is powerful in improving dim objects of interest against a bright foundation. By applying these transforms, it is feasible to disconnect significant subtleties in a picture and work on its general quality and handiness.

Thresholding techniques

Thresholding is a strong method in OpenCV utilized for sectioning pictures. It allocates pixel values in light of the threshold value provided, where every pixel is contrasted with the threshold value. In the event that the pixel value is more modest than the threshold, it is set to 0, in any case, it is set to a greatest value (by and large 255). Thresholding is usually utilized for isolating an item thought to be as a foreground from its background.

In computer vision, this strategy is usually performed on grayscale pictures. Accordingly, it means quite a bit to switch the picture over completely to grayscale prior to applying thresholding.

The essential Thresholding method is Binary Thresholding, where an threshold value is applied to every pixel in the picture. On the off chance that the pixel value is more less than the threshold, it is set to 0, in any case, it is set to a most extreme value. There are a few Straightforward Thresholding Procedures, including:
- cv2.THRESH_BINARY: sets the pixel worth to 255 assuming it is more prominent than the edge, else to 0 (dark)
- cv2.THRESH_BINARY_INV: rearranges the result of cv2.THRESH_BINARY
- cv2.THRESH_TRUNC: shortens the pixel power to the edge esteem
- cv2.THRESH_TOZERO: sets the pixel power to 0 for every one of the pixels with power not exactly the limit esteem

- cv2.THRESH_TOZERO_INV: modifies the result of cv2.THRESH_TOZERO

Another Thresholding strategy is Adaptive Thresholding, where the threshold value is determined for more smaller districts. This prompts different edge values for various locales concerning the adjustment of lighting. This method is valuable in situations where the lighting conditions change across the picture. The cv2.adaptiveThreshold capability is utilized for Versatile Thresholding, with two distinct techniques:

- cv2.ADAPTIVE_THRESH_MEAN_C: limit esteem is the mean of the blockSize×blockSize neighborhood of a pixel less a consistent worth
- cv2.ADAPTIVE_THRESH_GAUSSIAN_C: edge esteem is a weighted amount of the blockSize×blockSize neighborhood of a pixel less a steady worth.

In Otsu Thresholding, a threshold value is resolved consequently by examining the histogram of the picture. This method is helpful when the picture is bimodal, and that actually intends that there are two particular picture values. The cv2.threshold capability is utilized with the cv2.THRESH_OTSU banner to perform Otsu Thresholding. The threshold value is resolved naturally by tracking down a value that lies in the two tops in the histogram.

Generally, Thresholding is an integral asset that is utilized to portion pictures in view of the pixel values. It has numerous applications in Computer Vision, including object identification, background deduction, from there, the sky is the limit.

Summary

In summary, the algorithm captures an image of a slide label, analyzes its orientation, applies OCR and barcode detection modules, and then moves to the next steps of the pipeline, depending on the results of the previous steps. The algorithm focuses on the label region, enhances it through various image processing operations, and then computes the pixel intensities to decide whether the label is empty or not.

System Design Procedure

Python was used as the programming language to develop this feature. For all the processing steps, python libraries such as cv2, numpy, paddle OCR, pyzbar, pylibdmtx, pyzxing, and other such were used. The scanner and robot are controlled using the in-house developed Pramana Interface. The scanners and the robots have high end CPUs attached to them with linux OS. The slide basket and the drop baskets are all 3D printed and everything is mounted on a cluster assembly unit. The slides are filled in a basket and the robot feeds slides from the basket to the scanner, and the scanner then captures the images and scans the slide to produce dicom data. All the data from the scanner is stored in cluster NAS storage and can be further analysed on the Pramana UI or used as input to other prediction algorithms.

Pseudocode

1. Accept input as a picture of the slide label
2. Apply OCR on the picture to produce a text list with certainty values of each
3. Localize and decode any barcode tag tracked down in the picture
4. If the barcode decoded value decoded isn't empty, mark the picture as a non-flipped slide and send for examining.
5. In the event that no barcode tag value is decoded, perform flip procedure on the picture to make a mirror representation
6. Apply a progression of image processing steps, for example, obscuring, normal thresholding, adaptive thresholding, and denoising on the mirror representation
7. Apply OCR on the processed mirror image to create a word list with certainty values of each
8. Pass the two lists of words and confidence values, and calculate the total number of text boxes, absolute number of above threshold boxes, and length of high certainty words (for cases with just 1 or 2 text boxes on the label)
9. Make a decision in view of the circumstances:

a. If no or hardly any boxes are tracked down on the name, mark it as an ambiguous case

b. In the event that the two variants have an equivalent number of high certainty boxes, mark those names as questionable also

c. On the off chance that the first picture has all the more high certainty words, mark the picture as not flipped, and send it for examining

d. If the mirror representation has all the more high certainty words, mark the picture as reflected and keep it from examination

10. Assuming the label is ambiguous, move to the following stage of the pipeline
11. Apply image processing steps to decide whether the label is vacant or not
12. Crop the picture to focus on the label region
13. Apply blackhat operation to eliminate dull shadows of the apparatus parts on the edges
14. Apply normal thresholding and adaptive thresholding to the label picture
15. Apply morphological closing operation to deal with the noise on the picture
16. Register dark pixel intensities on both thresholded pictures
17. Compare dark pixel intensities with a threshold which was processed through several tests
18. If the dark pixel intensities are higher than threshold, mark the label as normal and send it for scanning
19. On the off chance that the dark pixel intensities are lower than limit, mark the label as blank or flipped, and keep it from sending it for examining

3.2 Diagrams for Analysis

1) Pipeline Structure

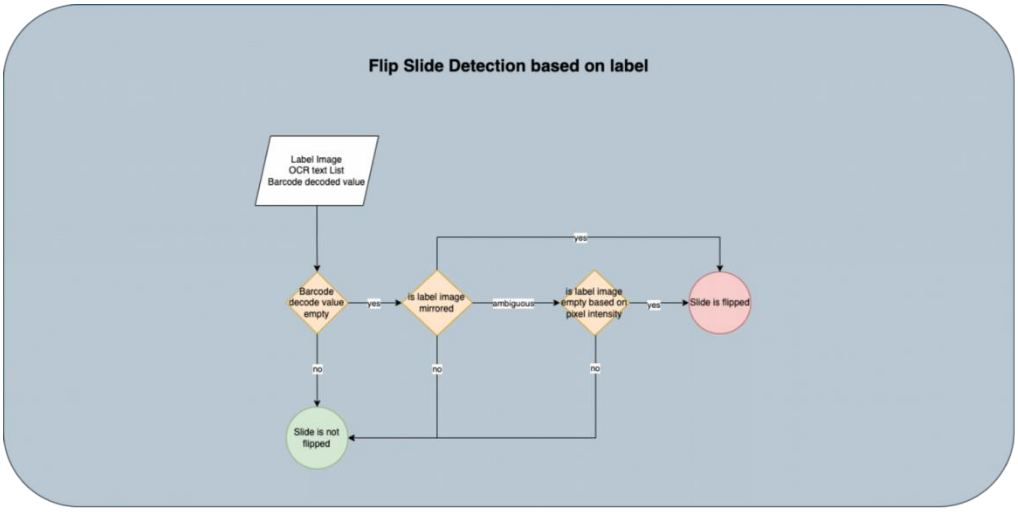


Diagram 1: Flip slide detection algorithm flowchart

2) Mirror Label Detection

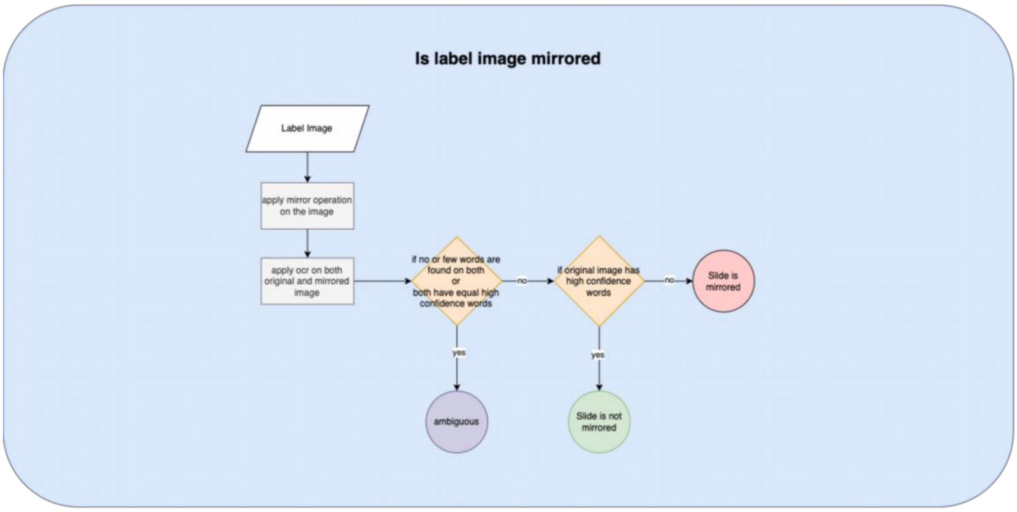


Diagram 2: Mirror label detection algorithm flowchart

3) Blank Label Detection

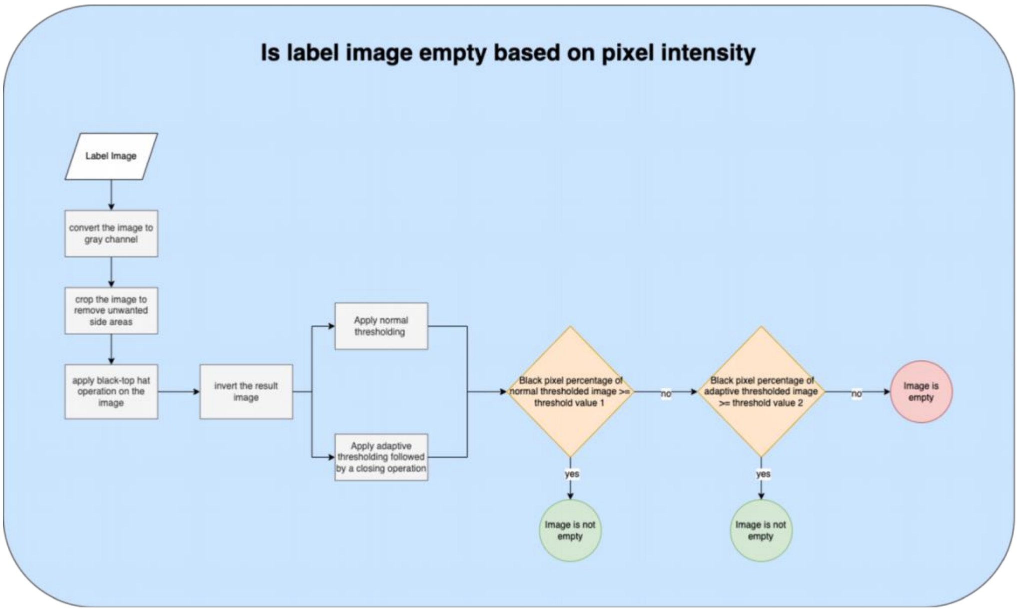


Diagram 3: Blank label detection algorithm flowchart

Chapter 04: Performance Analysis

4.1 System Properties

The system is designed in such a way that a very minimal extra time is spent for detection of the status of the label being flipped or not. The system uses robust and lightweight modules such as Paddle OCR, and all the barcode decoding libraries. Furthermore, localized cropped images are used to reduce analysis time even more and produce highly accurate results.

On an average it was noticed that this feature takes less than 0.17 seconds to detect the orientation. Even the results produced from this pipeline results better than certain machine learning models, against which it was tested. This pipeline produced far better results than a logistic regression model trained to detect the same. For the same purpose, mobilenet and vgg16 models were also analyzed for time and accuracy. Although they produced accurate results, they were expensive in terms of time.

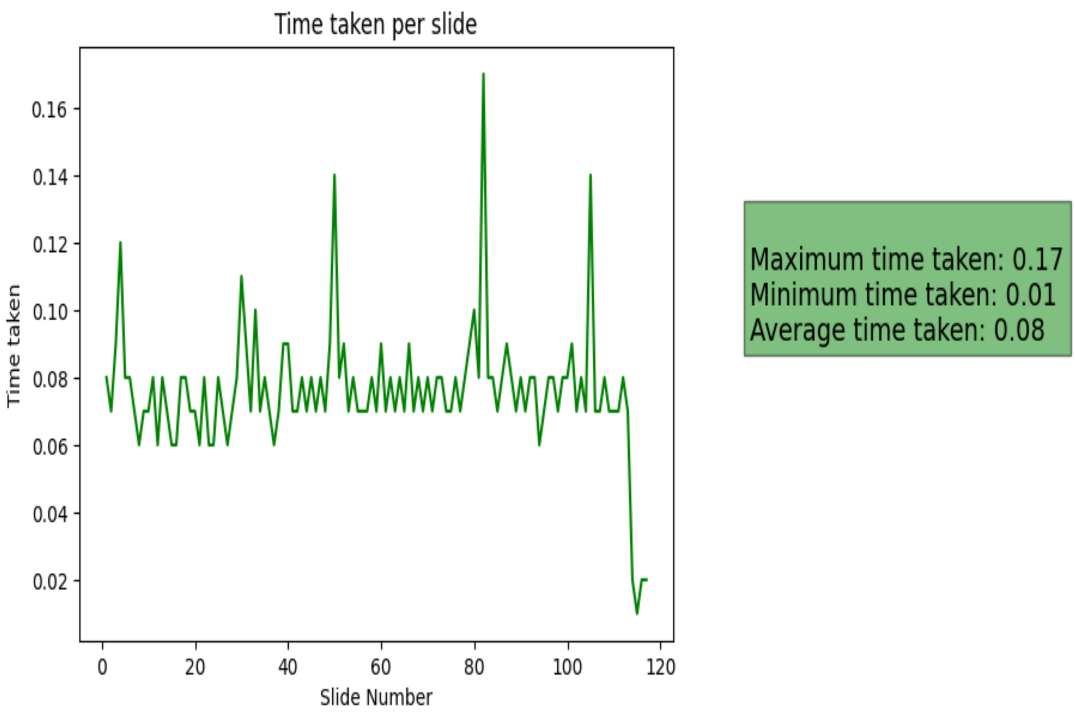


Fig. 5: Time taken per slide for the feature.

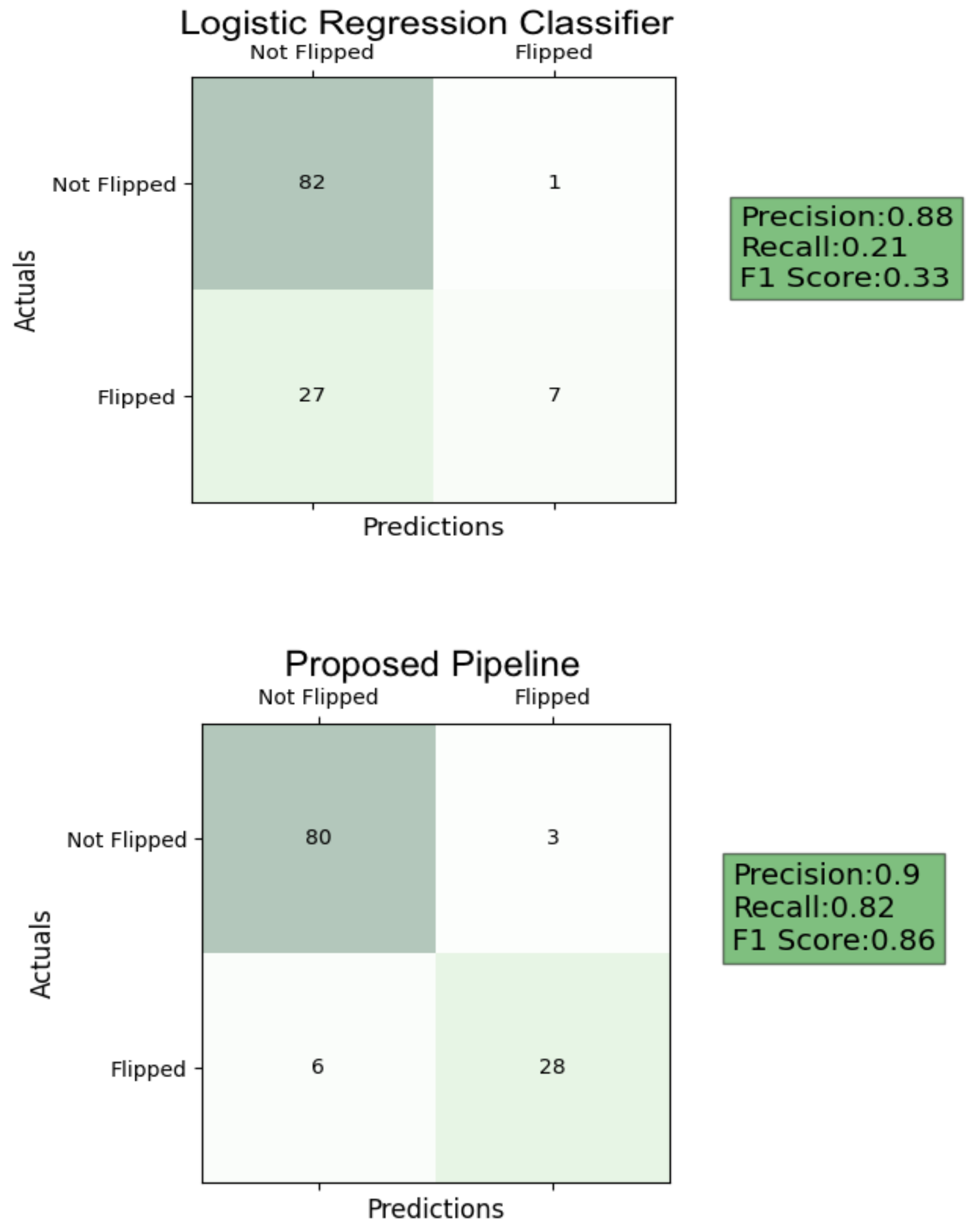


Fig. 4: Comparison of logistic regression classifier and the proposed pipeline over a sample dataset.

4.2 Output

Fig. 6

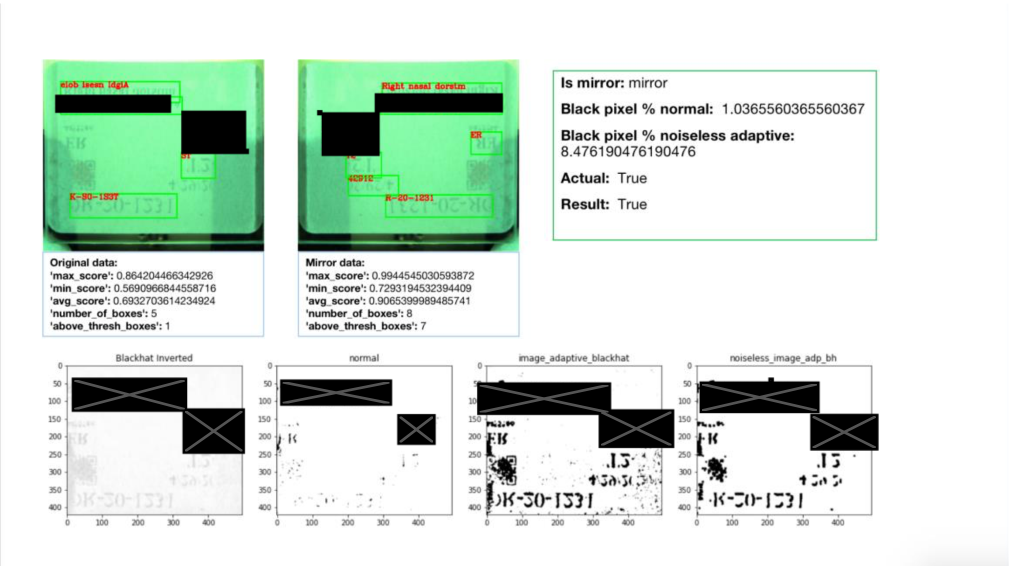


Fig. 7

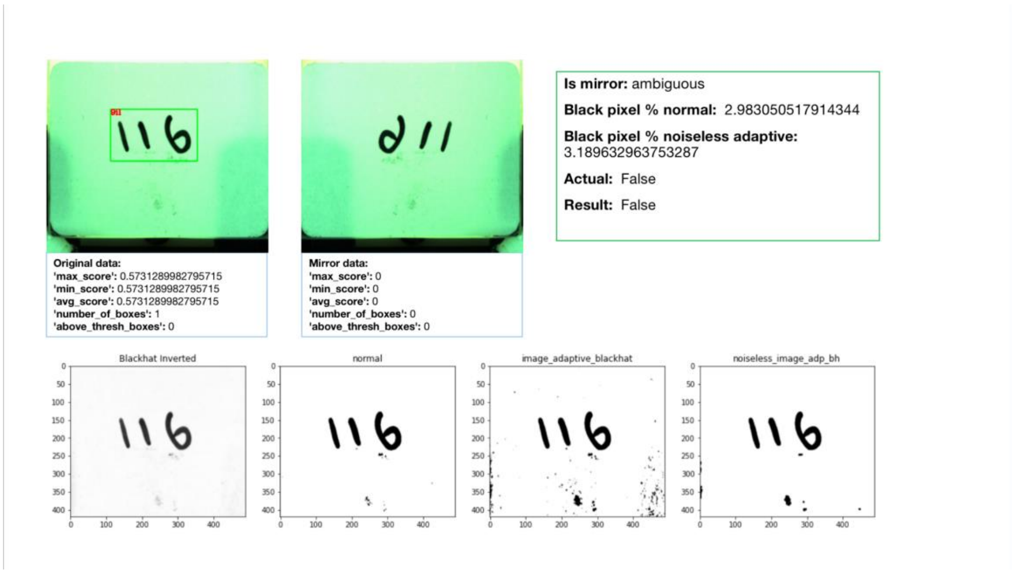

Is mirror: ambiguous
Black pixel % normal: 2.983050517914344
Black pixel % noiseless adaptive: 3.189632963753287
Actual: False
Result: False
Original data:
'max_score': 0.5731289982795715
'min_score': 0.5731289982795715
'avg_score': 0.5731289982795715
'number_of_boxes': 1
'above_thresh_boxes': 0
Mirror data:
'max_score': 0
'min_score': 0
'avg_score': 0
'number_of_boxes': 0
'above_thresh_boxes': 0
Blackhat Inverted
normal
image_adaptive_blackhat
noiseless_image_adp_bh

Fig. 8

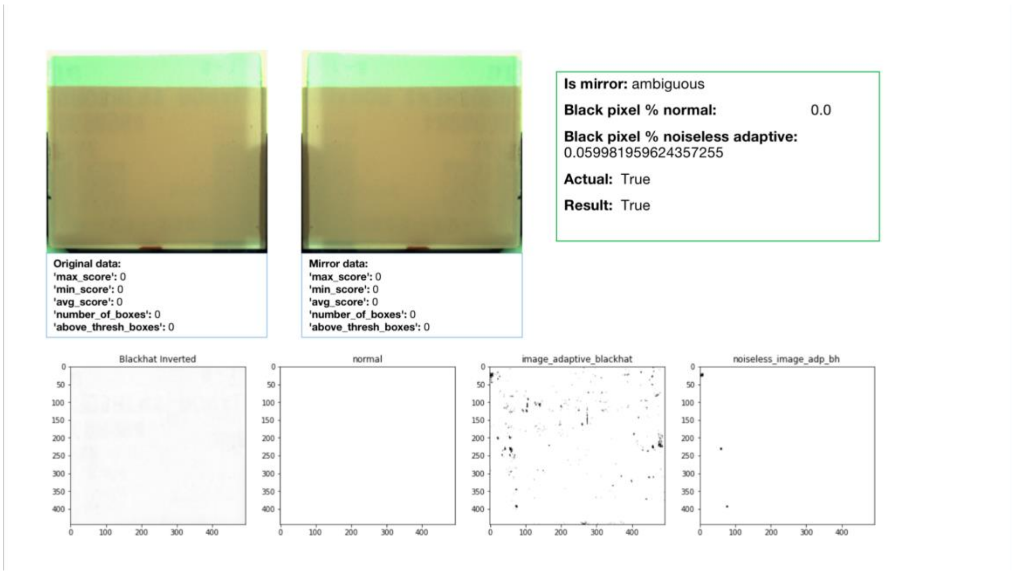
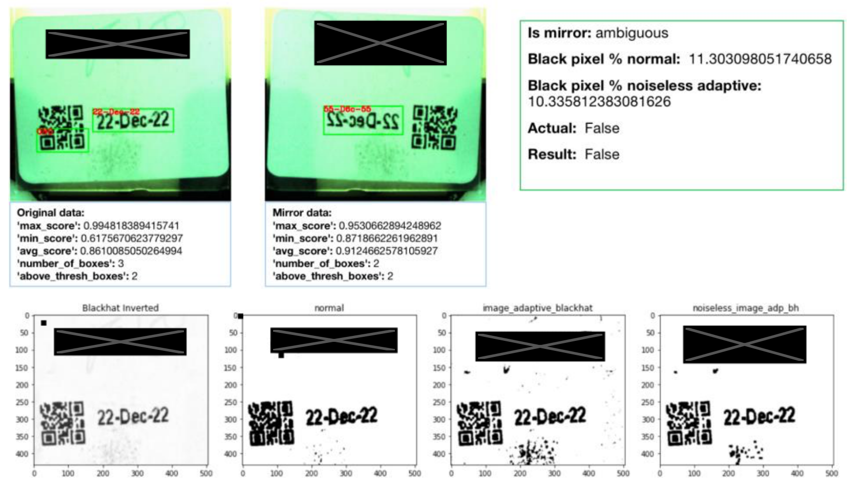
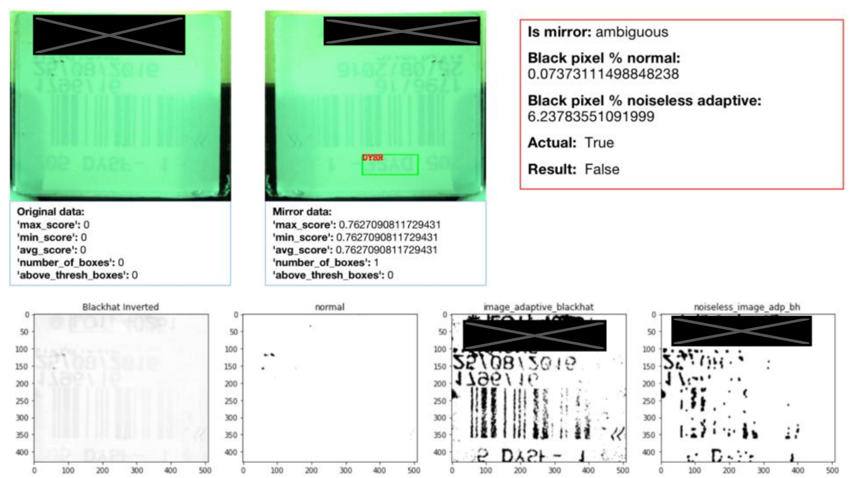

Is mirror: ambiguous
Black pixel % normal: 0.0
Black pixel % noiseless adaptive: 0.059981959624357255
Actual: True
Result: True
Original data:
'max_score': 0
'min_score': 0
'avg_score': 0
'number_of_boxes': 0
'above_thresh_boxes': 0
Mirror data:
'max_score': 0
'min_score': 0
'avg_score': 0
'number_of_boxes': 0
'above_thresh_boxes': 0
Blackhat Inverted
normal
image_adaptive_blackhat
noiseless_image_adp_bh

Fig. 9



Is mirror: ambiguous
Black pixel % normal: 11.303098051740658
Black pixel % noiseless adaptive: 10.335812383081626
Actual: False
Result: False
Original data:
'max_score': 0.994818389415741
'min_score': 0.6175670623779297
'avg_score': 0.8610085050264994
'number_of_boxes': 3
'above_thresh_boxes': 2
Mirror data:
'max_score': 0.9530662894248962
'min_score': 0.8718662261962891
'avg_score': 0.9124662578105927
'number_of_boxes': 2
'above_thresh_boxes': 2
Blackhat Inverted
normal
image_adaptive_blackhat
noiseless_image_adp_bh

Fig. 10



Is mirror: ambiguous
Black pixel % normal: 0.07373111498848238
Black pixel % noiseless adaptive: 6.23783551091999
Actual: True
Result: False
Original data:
'max_score': 0
'min_score': 0
'avg_score': 0
'number_of_boxes': 0
'above_thresh_boxes': 0
Mirror data:
'max_score': 0.7627090811729431
'min_score': 0.7627090811729431
'avg_score': 0.7627090811729431
'number_of_boxes': 1
'above_thresh_boxes': 0
Blackhat Inverted
normal
image_adaptive_blackhat
noiseless_image_adp_bh

Fig. 11

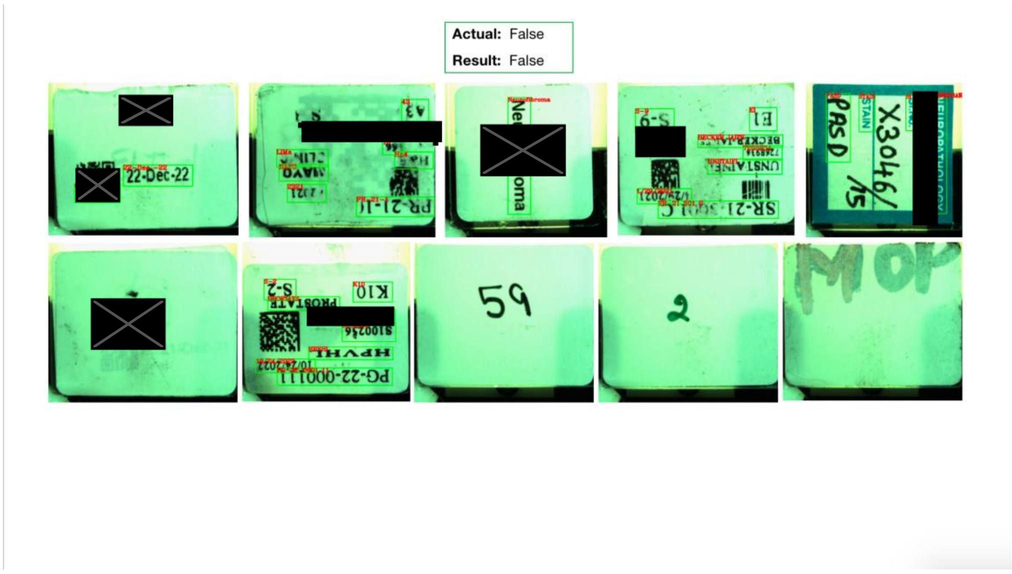


Fig. 12

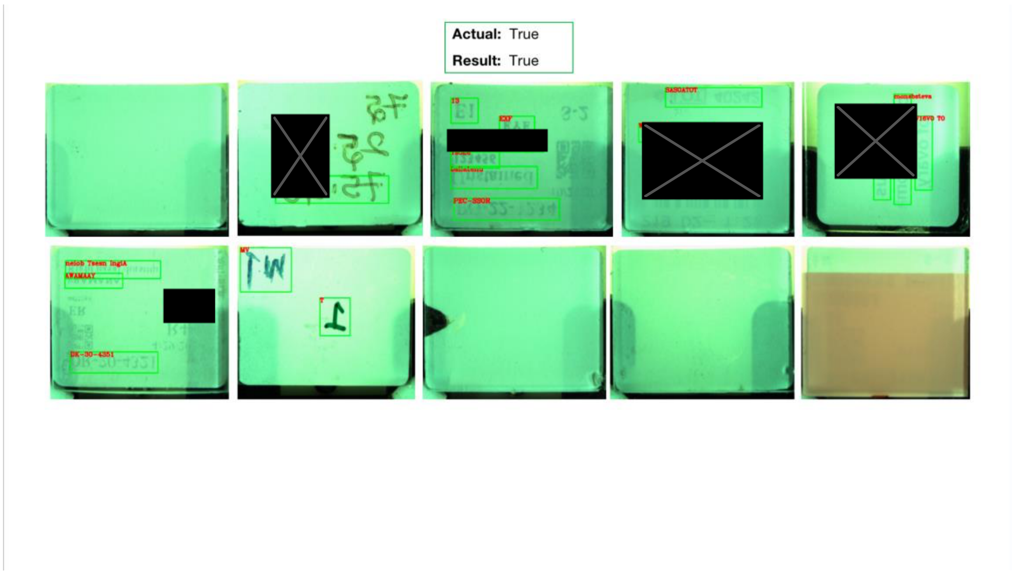


Fig. 13

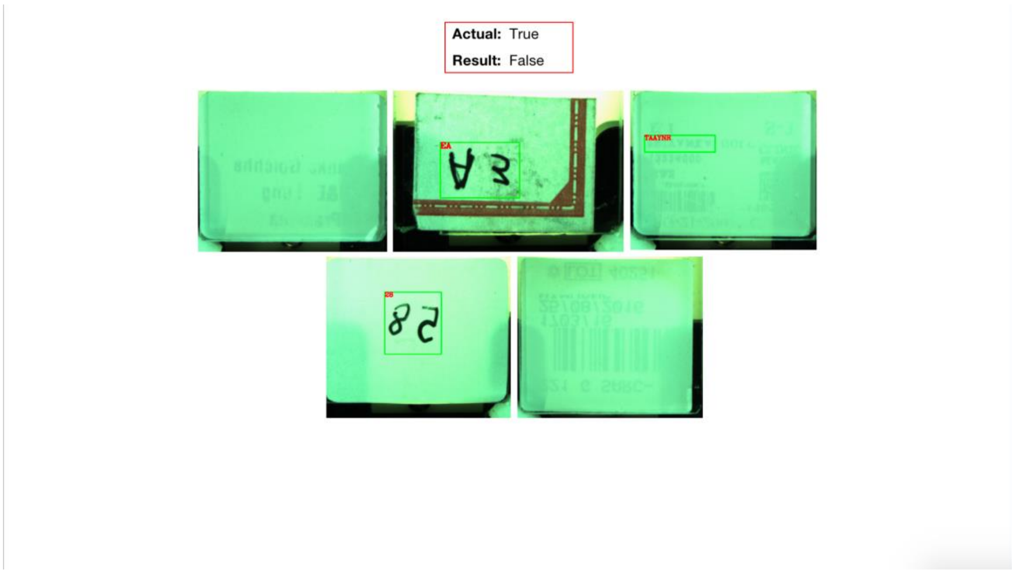


Fig. 14

Chapter 05: Results & Conclusions

5.1 Conclusion

A system with low computational time and high accuracy is achieved to detect whether the slide label is flipped or not. This system helps to reduce the scanning of incorrect data by preventing incorrect glass slides being fed for scanning. This also reduces time wastage and improves the overall performance of the complete cluster. The data provided to the algorithms in the next stage is also correct and thus the final analysis reports are also even more accurate.

This algorithm successfully identifies normal labels, flipped labels and blank labels as well. Only the slides which the algorithm is confident about being flipped are prevented and rest are passed to scan, in order to capture as much as medical data can be captured. At times when there is an ambiguity then for the sake of belief that the user must have kept it correctly, we pass the slide to scan so that we might not miss a possibly correct slide.

5.2 Future Scope

The algorithm still has certain ambiguous cases where it fails, and can be worked upon those. Certain labels have text or annotations on both sides, making it difficult to judge its orientation. Certain labels have scratches or other kinds of unwanted marks, and again result in an ambiguity. There are some annotations or handwritten text on the glass part of the slide, and which is visible from behind the label, which is also a confusing case for the algorithm. All such cases can be dealt with the proper use of distinct features these cases carry with respect to the normal labels.

5.3 Application

Clinical research centers: The framework can be utilized in clinical labs for robotized label reading and data entry. This can reduce mistakes and work on the effectiveness of the testing system.

Operations and warehousing: The framework can be utilized in coordinated factors and warehousing enterprises to automate inventory management. Names on items and bundles can be perused and placed into an information base consequently, lessening the time and exertion expected for manual data entry.

Manufacturing: The system can be used in manufacturing industries to automate quality control processes. Labels on products and parts can be read automatically, and any defects can be identified and flagged for further inspection.

Libraries and chronicles: The framework can be utilized in libraries and files for automated classifying of books, reports, and different materials. Names can be read and placed into a database consequently, making it simpler to look and recover materials.

Retail: The framework can be utilized in retail locations for automated value checking and inventory administration. Marks on items can be read and placed into a dataset automatically, and costs can be refreshed progressively.

Farming: The framework can be utilized in horticulture for automated tracking of harvests and animals. Labels on plants and creatures can be read and placed into a dataset automatically, making it more straightforward to track development and health data.

References

1) "A Study on Detection of Flipped Images in Text Data," by Seung-Ki Ryu et al. (2018).

2) "Automatic Detection of Mirror Images in Medical Images," by Sushma Guduru et al. (2016).

3) "Mirror Image Detection Using Convolutional Neural Networks," by Ryohei Hisano et al. (2018).

4) "Detection of Mirrored Text in Images," by Tarun Pruthi et al. (2018).

5) "A Novel Approach to Mirror Image Detection in Printed Documents,"

6) OpenCV documentation - https://docs.opencv.org/4.x/

7) Types of morphological operations on images - https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html

Appendices

Code

```
def processing_with_black_hat(image):
    ksize = 65
    rectKernel = cv2.getStructuringElement(cv2.MORPH_RECT, (ksize, ksize))
    blackhat = cv2.morphologyEx(image, cv2.MORPH_BLACKHAT, rectKernel)
    # inverting image
    image_inv = cv2.bitwise_not(blackhat)
    return(image_inv)
```

```
def get_black_pixel_percentage(gray_image):
        perc_black_pixels_freq = 0
        try:
            img = gray_image.copy()
            img_shape=img.shape
            total_pixels = 1
            for i in range(0, len(img_shape)):
                total_pixels = total_pixels*img_shape[i]

            zeros=(img==0).sum()

            perc_black_pixels_freq = (zeros/total_pixels)*100

        except Exception as e:
            print("Excepton in get_black_pixel_percentage : ", e)

        return(perc_black_pixels_freq)
```

```
def get_black_pixel_percentage(gray_image):
    perc_black_pixels_freq = 0
    img = gray_image.copy()
    img_shape=img.shape
    total_pixels = 1
    for i in range(0, len(img_shape)):
        total_pixels = total_pixels*img_shape[i]

    zeros=(img==0).sum()

    perc_black_pixels_freq = (zeros/total_pixels)*100

    return(perc_black_pixels_freq)
```

```
def pixel_entropy_process(tiles,channel=0):
    for tile in tiles:
        processed_img = tile.processed[channel]
        black_pixel_perc_org = get_black_pixel_percentage(processed_img)
        tile.black_perc = black_pixel_perc_org
        if(black_pixel_perc_org >= tile.black_frequency):
            tile.assign_status(False)
        else:
            tile.assign_status(True)
```

```
# |---------------------------------------------------------------------------|
# get_mirror_image returns images after mirroring them
# @param org_image_path - original color image path
# @param mirror_image_path - path where to save mirrored image
# @return mirror_image_path mirrored image path
# |---------------------------------------------------------------------------|
def get_mirror_image(self, org_image_path, mirror_image_path):
    org_image = cv2.imread(org_image_path)
    mirror_image = numpy.fliplr(org_image)
    cv2.imwrite(mirror_image_path, mirror_image)
    return mirror_image_path
```

```python
# |---------------------------------------------------------------------------|
# get_avg_max_min returns average, minimum, maximum of score and number of boxes
# @param retCode for status of OCR
# @param retDict for values of OCR
# @return dictionary data with avg_score, max_score, min_score, number_of_boxes
# |---------------------------------------------------------------------------|
def get_avg_max_min(self, retCode, retDict):
    data = {
        "avg_score": 0,
        "max_score": 0,
        "min_score": 0,
        "number_of_boxes": 0,
    }
    def avg(arr):
        if len(arr) == 0:
            return 0
        return sum(arr)/len(arr)

    def maximum(arr):
        if len(arr) == 0:
            return 0
        return max(arr)

    def minimum(arr):
        if len(arr) == 0:
            return 0
        return min(arr)
    if retCode:
        data["avg_score"]= avg(retDict["scores"])
        data["max_score"]= maximum(retDict["scores"])
        data["min_score"]= minimum(retDict["scores"])
        data["number_of_boxes"]= len(retDict["scores"])

    return data
```

```python
img_path = item[0]
image, image_cropped, blackhat, normal_processed, image_rect = image_preprocessing(img_path)
window_size = 199
image_adaptive_blackhat = cv2.adaptiveThreshold(blackhat, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, wi

kernelSize = (5,5)                                          (constant) MORPH_CLOSE: int
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, kernelSize)
noiseless_image_adp_bh = cv2.morphologyEx(image_adaptive_blackhat, cv2.MORPH_CLOSE, kernel)

mirror_image = numpy.fliplr(image_cropped)
show_tiles(image_rect, "Original",image_cropped,"image_cropped", mirror_image, "mirror_image")
show_tiles(blackhat, "Blackhat Inverted", normal_processed, "normal", image_adaptive_blackhat, "image_adaptive_black

bpp_norm = get_black_pixel_percentage(normal_processed)
bpp_adp = get_black_pixel_percentage(image_adaptive_blackhat)
bpp_noiseless_adp_bh = get_black_pixel_percentage(noiseless_image_adp_bh)
```

Charts and Flowchart