Onboarding Customers in Car Sharing Systems: Implementation of Know Your Customer Solutions

Magzhan Kairanbay[a]
Array Innovation, Bahrain

Keywords: Know Your Customer, Object Detection, Optical Character Recognition.

Abstract: Car-sharing systems have become an essential part of modern life, with Know Your Customer (KYC) processes being crucial for onboarding users. This research presents a streamlined KYC solution designed to efficiently onboard customers by extracting key information from identity cards and driving licenses. We employ techniques from Computer Vision and Machine Learning, including object detection and Optical Character Recognition (OCR), to facilitate this process. The paper concludes by exploring additional features, such as gender recognition, age prediction, and liveness detection, which can further enhance the KYC system.

1 INTRODUCTION

The proliferation of vehicles has been linked to a wide array of environmental and economic challenges. Increased emissions from automobiles contribute significantly to air quality degradation, which poses serious health risks to the population (Cerovsky and Mindl, 2008). Furthermore, traffic congestion adversely impacts national economic performance (Kagawa et al., 2013). In response to these issues, governments are actively seeking strategies to mitigate car usage.

One promising solution is the implementation of car sharing systems. Car sharing can take various forms; while some people share rides to a common destination, others may utilize a vehicle when its owner is not actively using it. This paper focuses on the latter model, in which individuals can access vehicles for personal use through a shared platform.

Car-sharing systems can be categorized into two primary approaches: traditional car-sharing, which operates a fleet of vehicles owned by the company, and peer-to-peer (P2P) car-sharing, where individuals offer their own vehicles for community use. The P2P model typically requires fewer resources from the company, making it an attractive option for both service providers and users.

A critical component of any car-sharing system is the onboarding process, which ensures that customers can legally operate vehicles within the system. Effective onboarding relies on robust customer identification processes that gather essential information, including full name, IC number, driving license number, and photographic verification. These data are vital to establish trust and enable the seamless operation of car-sharing services.

In this paper, we present a comprehensive approach to implementing a Know Your Customer (KYC) solution tailored for car-sharing systems. Our proposed system integrates several machine learning (ML) models to facilitate the following functionalities:

- Face detection
- Face comparison
- Identity card (IC) detection
- Driving license detection
- Verification of the authenticity of ICs, driving licenses
- Extraction of Regions of Interest (ROI)
- Optical Character Recognition (OCR) for extracted ROIs
- User communication with the ML solution

Fig. 1 illustrates the architecture of the proposed system, showcasing its key components and their interactions.

[a] https://orcid.org/0000-0002-8741-434X

Kairanbay, M.
Onboarding Customers in Car Sharing Systems: Implementation of Know Your Customer Solutions.
DOI: 10.5220/0013145100003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 3: VISAPP*, pages 381-391
ISBN: 978-989-758-728-3; ISSN: 2184-4321

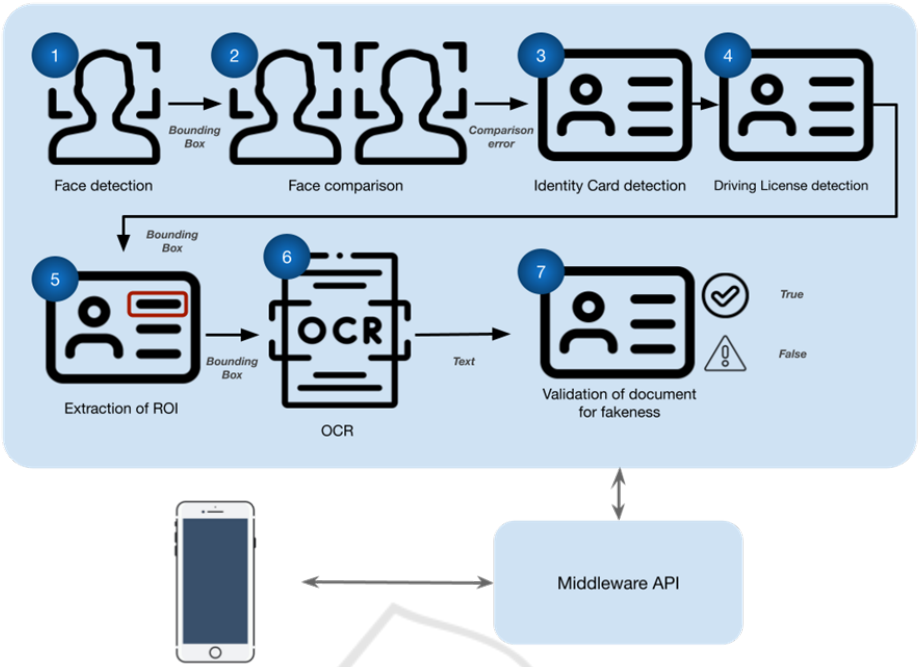

Figure 1: ML-based KYC Solution Architecture for Car Sharing Systems.

2 RELATED WORK

KYC systems are essential for financial institutions, playing a key role in preventing illegal activities such as money laundering, fraud, and terrorist financing. These systems verify client identities, ensure compliance with regulatory requirements, and mitigate financial crimes. Traditionally, KYC processes involved manual steps; however, advancements in technology have led to the development of automated solutions aimed at increasing efficiency and accuracy. This shift has been driven by the need for quicker and more secure customer onboarding while ensuring compliance with rigorous regulatory standards.

Recent developments in KYC technologies include the adoption of biometric authentication (Miller and Smith, 2019), ML (Charoenwong, 2023), and blockchain (Liu and Zhang, 2019). Biometric methods, such as facial recognition (Jain and Gupta, 2017) and fingerprint scanning (Al and Kumar, 2020), have significantly enhanced the accuracy and security of identity verification procedures. ML techniques, including anomaly detection (Bhardwaj and Sharma, 2021) and pattern recognition (Yoon and Kim, 2020), help identify suspicious activities in real time. Additionally, blockchain technology (Zohdy and Thomas, 2018) provides decentralized solutions for secure and transparent sharing of customer data. However, despite these technological advancements, challenges such as regulatory compliance, data privacy concerns, and the financial costs associated with implementing sophisticated systems remain significant issues for financial institutions (Liu and Zhang, 2019).

The increasing importance of KYC systems is not limited to financial institutions alone but spans various industries. While the documents required for verification may vary based on the industry (e.g., finance, healthcare, or real estate), the underlying principles of document validation and data extraction remain the same. This section reviews current KYC solutions, examining their strengths and limitations, and outlines how proposed innovations aim to address these challenges.

Recent studies highlight the application of ML techniques in improving KYC processes. For instance, research by (ACTICO, 2023) shows that supervised learning methods, particularly Random Forests, are effective in enhancing the performance of KYC applications. Their findings suggest that adopting these techniques can reduce the number of clarification requests by up to 57%, significantly streamlining the compliance process and improving efficiency.

(Technologies, 2023) employed several advanced techniques to tackle the KYC challenge, including:

- Convolutional Neural Networks (CNNs) utilizing Python and TensorFlow
- OpenCV for computer vision tasks

- Optical Character Recognition (OCR) and Machine Readable Zone (MRZ) packages

Their approach involves scanning documents to extract personal information and passport expiration dates. Once extracted, this data is compared with database records to validate the submitted documents. Their model classifies documents as verified, expired, canceled, or mismatched based on validation results. Documents that cannot be confidently categorized are referred for manual review. Over time, the model benefits from manual classifications through automated retraining, integrating new and corrected data. The authors assert that ML-based solutions can significantly expedite the customer onboarding process, achieving document verification in one-tenth the time required for manual processing. Furthermore, their solution boasts high accuracy and efficiency, adhering to standard procedures while minimizing manual intervention. They report a 70% reduction in manual effort for KYC verification, along with a 70% improvement in resource management, allowing organizations to redirect human resources to more value-added tasks.

Conversely, (Charoenwong, 2023) argue that KYC solutions often do not yield significant benefits for banks. They contend that the challenges associated with KYC are not merely issues of data science or ML, but rather a stem from systemic incentives within the banking industry. They posit that the underlying problems are trivial, yet banks may profit from circumventing these processes.

In the following section, we present our solution for addressing the KYC challenge through deep learning methods combined with OCR techniques. The literature reviewed indicates that customer onboarding processes can be automated and accelerated. This research aims to validate that premise, demonstrating that ML approaches can effectively automate and expedite the onboarding process with minimal effort. We also aim to showcase the applicability of such solutions in real-world scenarios, including car-sharing systems. To substantiate our hypothesis, we conducted experiments using Kazakhstani documents, specifically ICs and driving licenses. These documents were processed using object detection and character recognition models, with the details of our proposed solution outlined below.

3 PROPOSED ML-BASED KYC SYSTEM

The proposed solution encompasses several key components, including:

- Face detection
- Face comparison
- IC detection
- Driving license detection
- Validation of ICs and driving licenses for authenticity
- Extraction of regions of interest (ROI)
- Optical character recognition (OCR) for the extracted regions

In the following subsections, we will explore each component in detail. Our solution is designed for real-world applications, streamlining the customer onboarding process, reducing manual workload, and enhancing customer satisfaction while accommodating a greater number of users.

3.1 Data Collection and Labeling

ML methodologies rely on data-driven approaches, necessitating the collection of data prior to initiating any training processes. Given that we are employing a supervised learning strategy, it is essential for the data to include corresponding ground truth labels. Our focus will primarily be on object detection tasks, such as identifying ICs, driving licenses, and identity numbers, where the goal is to locate specific objects within the provided images. Object detection involves drawing the smallest bounding box that encapsulates the object of interest. For this bounding box, we identify the coordinates of the top-left and bottom-right corners.

For each object detection task, we have gathered 2,000 images, and for each image, we have manually annotated the bounding boxes. Figure 2 below illustrates the data alongside its corresponding bounding box. Each bounding box serves as a ground truth label, defined by the coordinates $(x_{\text{topleft}}, y_{\text{topleft}})$ and $(x_{\text{bottomright}}, y_{\text{bottomright}})$. There are no constraints on the input image sizes, allowing for the use of images with any dimensions.

3.2 Face Detection

Face detection is a fundamental component of most KYC systems (Pic et al., 2019), (Darapaneni et al., 2020), (Do et al., 2021). Since many official documents include a photo of the customer, it is crucial to verify that the individual in the photo matches the person during the onboarding process. To achieve this, our proposed solution requires customers to upload a selfie while holding their IC. This allows us to detect faces from both the selfie and the IC, enabling a

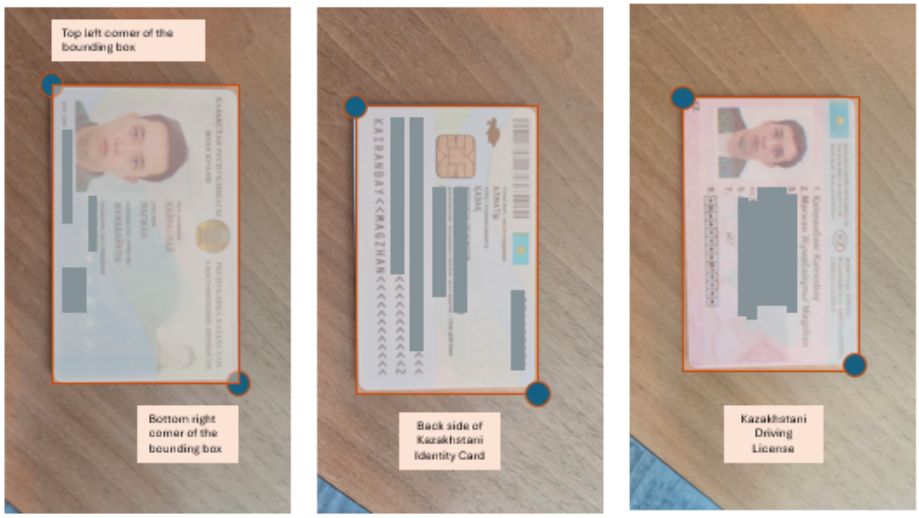

Figure 2: Sample data for a) front and b) back side of IC and c) driving license detection.

Figure 3: Face detection from the selfie and IC.

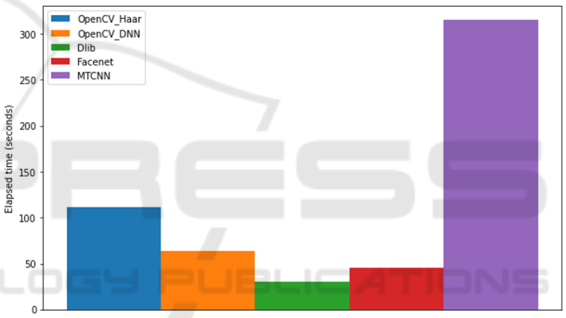

Figure 4: Comparison of face detection models based on processing time.

comparison to confirm that they belong to the same individual (see Fig. 3).

Our face detection implementation utilizes a variety of algorithms (Sun et al., 2018), (Mita et al., 2005), (Chang-Yeon, 2008), (Dalal and Triggs, 2005). (Stefanovic, 2023) conducted a comparative analysis of five face detection algorithms: Haar cascade, OpenCV DNN, Dlib, MTCNN, and Facenet. Their results indicated that while the detection accuracy of these models is comparable (Table 1), Dlib stands out for its processing speed, completing image processing in approximately 30 seconds, compared to Facenet and OpenCV DNN, which take about 50 seconds, Haar cascade around 100 seconds, and MTCNN approximately 300 seconds (Stefanovic, 2023) (see Fig. 4). Dlib is integrated into a user-friendly "face recognition" API, making it straightforward to implement in our solution. Given our priority of expediting customer onboarding, we opted to utilize this "face recognition" API (fac, 2024), which offers two methodologies: Histogram of Oriented Gradients (HoG) (Dalal and Triggs, 2005) and Convolutional Neural Network (CNN) (cnn, 2024).

Table 2 outlines the differences between these approaches, highlighting that HoG processes images 16 times faster than CNN. Since rapid onboarding is a primary requirement, we decided to use HoG, especially as the faces we detect will consistently be presented from a frontal angle, rendering additional CNN features unnecessary. The face extraction using HoG is subsequently classified with Linear Support Vector Machines.

During the face detection process, it is imperative to ensure that facial landmarks, such as the eyes, eyebrows, and lips, are visible. If any of these landmarks are obstructed, we cannot proceed to the next step. Common scenarios leading to landmark blockage include obscured lips due to mask-wearing, which has become prevalent during the COVID-19 pandemic. Therefore, we instruct users to remove masks when taking their photos. Other obstructions,

Table 1: Comparison of Face Detection Methods: Accuracy and Performance.

Method	Accuracy	Key Strengths	Performance
OpenCV Haar Cascades	90% - 95%	Fast, real-time detection, low computational requirements	Fast, but less accurate on diverse faces (e.g., varied poses)
OpenCV DNN (e.g., ResNet)	96% - 98%	Higher accuracy than Haar, robust to pose/occlusion/lighting	Slower than Haar, but real-time with proper hardware
Dlib (CNN-based)	95% - 98%	Accurate, works well with various poses and occlusions	Slower than Haar, but more accurate, especially with CNN
FaceNet	98% - 99%	Highly accurate, also provides embeddings for face recognition	Slower than other methods, requires high computational power
MTCNN	95% - 98%	Good for detecting faces at various angles and scales	Moderate speed, good accuracy, can be slower than Haar and Dlib

Table 2: Differences Between HoG and CNN-Based Solutions.

HoG	CNN
Detects faces primarily from frontal angles, making it less effective for identifying faces at various angles.	Capable of detecting faces from a wide range of angles.
Detection time is 0.2 seconds when using a CPU.	Detection time is 3.3 seconds when using a CPU.

such as glasses or long hair, can also hinder visibility of the eyebrows and eyes. To address these issues, we will provide a user manual outlining proper practices for capturing the selfie photo. Any photos that do not meet the visibility criteria for processing through our ML solutions will be subject to manual review. It is crucial that all requirements—visibility of lips, eyebrows, and eyes—are confirmed during the manual KYC validation process. Each case must ensure the presence of two faces: one from the selfie and one from the document. Once both faces are available, we can begin the comparison to ascertain if they belong to the same person. The subsequent subsection will detail the face comparison methodology.

3.3 Face Comparison

For each detected face, we need to obtain its face encodings, which are numerical representations of the face stored as a one-dimensional array. Typically, these encodings are derived from the penultimate fully connected layers of CNNs in deep learning models. However, since we are utilizing a HoG-based approach, we will focus on extracting HoG features. The process for obtaining these features is illustrated step-by-step in Figure 5.

The first step of the algorithm involves calculating the centered horizontal and vertical gradients, which can be expressed mathematically as

$$f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \frac{\partial f/\partial x}{\partial f/\partial y}, \tag{1}$$

where $g_x = \frac{\partial f}{\partial x}$ and $g_y = \frac{\partial f}{\partial y}$ represent the derivatives in the x and y directions. These derivatives can be approximated using convolution with the filters $h = [-1, 0, 1]$ and h^T in their respective directions. Once the image derivatives are calculated, they can be used to derive the gradient direction and magnitude, computed as

$$\theta = \tan^{-1}\left(\frac{g_y}{g_x}\right), \tag{2}$$

and

$$(g_x^2 + g_y^2)^{0.5}, \tag{3}$$

respectively.

Next, the image is divided into overlapping blocks of 16×16 pixels with a 50% overlap. Each block comprises 2×2 cells, each sized 8×8 pixels. We then quantize the gradient orientations into N bins and concatenate these to form the final feature vector. The "Face Recognition API" provides a method for obtaining these face encodings, which are represented as a one-dimensional array of length 128. For the i-th face encoding, we denote it as

$$f(i) = [f(i)_0, f(i)_1, f(i)_2, \ldots, f(i)_{127}].$$

Figure 6 illustrates the values of this feature vector, which range from -0.3 to 0.3.

Once we have the face encodings for two faces (face i and face j), we need to determine their similarity. To do this, we calculate the Euclidean distance between the two vectors (Recognition,). This distance measures the discrepancy between the vectors: if the distance is close to 0, the vectors are similar; if it

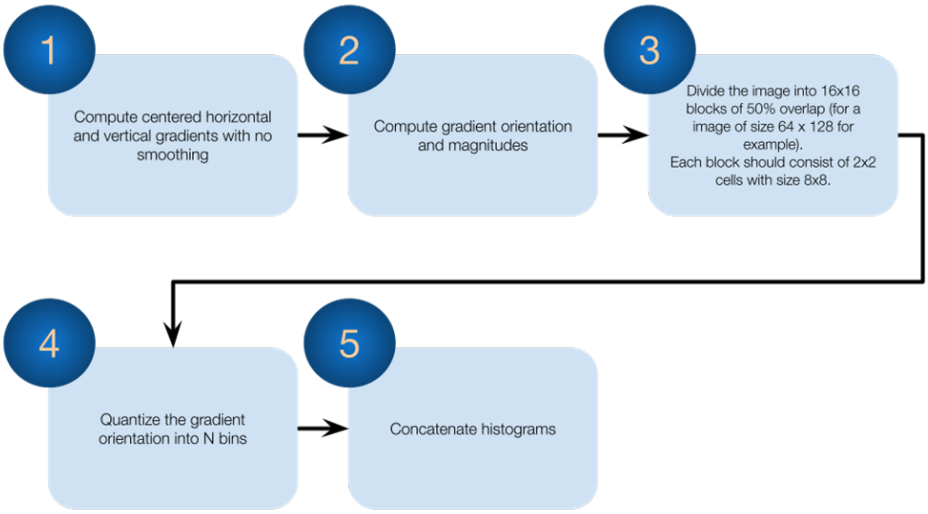

Figure 5: Feature extraction technique utilized by the HoG algorithm.

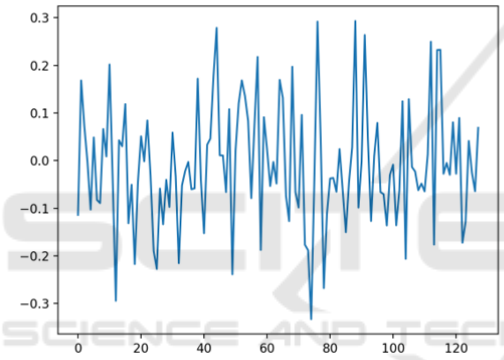

Figure 6: Visual representation of the feature vector for the "face recognition" API.

is significantly greater than 0, the vectors are dissimilar, indicating a low probability that the faces belong to the same person. We establish a threshold of 0.6 for this validation process. Thus, if the distance is below 0.6, we conclude that the faces (face i and face j) belong to the same person; otherwise, they belong to different individuals, as expressed in the equations below:

$$\text{dis}(i,j) = \left(\sum_{k=0}^{127}(f_k(i) - f_k(j))^2\right)^{0.5} < 0.6 \quad \text{(indicating the same person)}, \tag{1}$$

$$\text{dis}(i,j) = \left(\sum_{k=0}^{127}(f_k(i) - f_k(j))^2\right)^{0.5} \geq 0.6 \quad \text{(indicating different individuals)}, \tag{2}$$

Here, $\text{dis}(i,j)$ denotes the Euclidean distance between feature vectors i and j. While we are using the Euclidean distance for this purpose, other distance metrics could also be applied. According to the authors (fac, 2024), optimal performance is achieved with a threshold of 0.6, which we adopt in our approach. If the threshold value is lowered, the criteria for matching become stricter; conversely, raising the threshold makes the criteria more lenient.

3.4 IC Detection

After confirming that the faces in the selfie and IC belong to the same individual, we proceed to the next step: extracting key information from the documents. The IC serves as the primary document for verifying a person's identity. We will focus on retrieving the Identity Number from the card, as this is typically sufficient; additional details such as fines or penalties can be accessed using just the Identity Number.

There are two approaches to IC detection. The first involves developing a ML model specifically for detecting ICs. The second, simpler approach enhances detection accuracy by employing a bounding box drawn in the application. The user simply needs to align the IC with this box for effective detection and cropping. Therefore, the second approach is strongly recommended (Fig. 7). For the first method, we utilized an object detection technique known as Mask R-CNN (He et al., 2017). The architecture of Mask R-CNN is illustrated in Figure 8 (He et al., 2017), demonstrating its capability to segment the target object. Once the object is identified, we can establish the bounding box by determining the top-left and bottom-right coordinates. Mask R-CNN achieved an accuracy

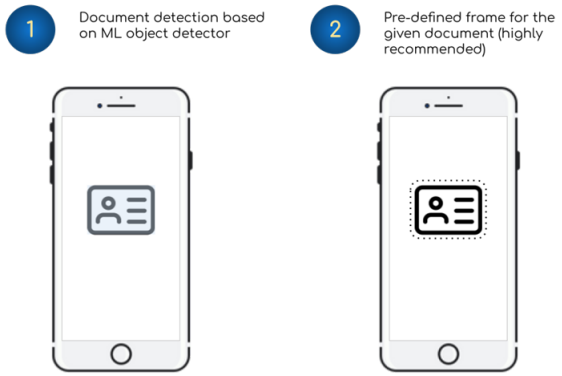

Figure 7: Two methods for document detection, with the second approach being strongly recommended.

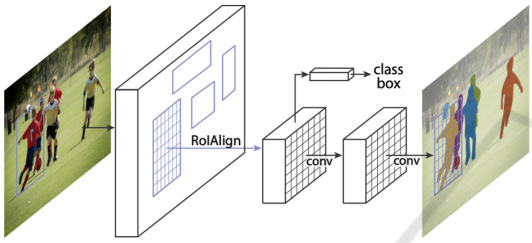

Figure 8: Architecture of the Mask R-CNN Algorithm.

of 91% for this task. We used the Jaccard index (Leydesdorff, 2008) as our evaluation metric, which is defined as the intersection over union of our predictions and the ground truth labels (see Figure 9).

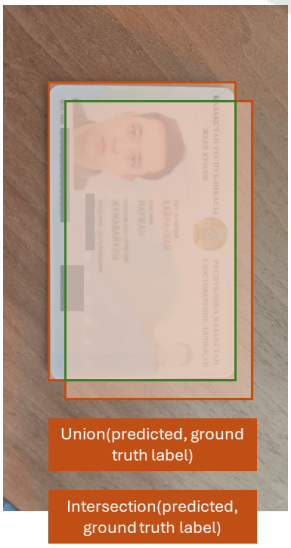

Figure 9: The intersection and union between the predicted bounding box and the ground truth bounding box.

$$\text{Jaccard}(\text{predicted}, \text{ground truth}) = \frac{\text{predicted} \cap \text{ground truth}}{\text{predicted} \cup \text{ground truth}} \quad (4)$$

In an ideal scenario where object detection is perfect, the Jaccard index equals 1, indicating complete overlap between the predicted and ground truth bounding boxes. Once the IC is detected, we will proceed to validate its authenticity. However, before that, we will discuss the detection of driving licenses.

3.5 Driving License Detection

Detecting a driving license follows a process similar to that of IC detection. We have two approaches available, with the second approach being strongly preferred. For the first approach, we will employ Mask R-CNN, which achieves an accuracy of 91% in detecting driving licenses (as shown in Table 3). We utilized nearly 2,000 data samples, allocating 80% for training and 20% for testing in both IC and driving license detection tasks. The accuracy for both tasks is comparable due to the consistent dataset and training/testing split used for each. The primary distinction lies in the specific document type being identified.

Table 3: Document Detection Accuracy.

Task	Accuracy
IC Detection	91%
Driving License Detection	91%

Once the necessary document is detected, we can proceed to extract key information from it. Typically, the expiration date and type of driving license are critical data points needed for customer onboarding. The extraction of regions of interest (ROIs) from these documents will be addressed in subsequent chapters. Before that, we will discuss the validation process for both ICs and driving licenses to ensure their authenticity.

3.6 Validation of IC and Driving License for Authenticity

After the documents have been identified, the subsequent step is to verify their authenticity. Individuals sometimes attempt to use counterfeit documents for various illicit purposes, including fraud, theft, vandalism, and other criminal activities related to vehicles. Therefore, it is essential to ensure that customers are presenting legitimate and accurate documents. The methods of validation differ depending on the type and format of the documents. Typically, these documents adhere to specific standards, which include

a b c

Figure 10: Verification of the document's authenticity by identifying key landmarks from a) front, b) back side of IC and from c) driving license.

identifiable landmarks located at designated points. Consequently, the initial validation process involves detecting these landmarks (Fig. 10). Once we confirm the presence of all required landmarks, this validation step is deemed complete. This method represents a basic form of validation, as the primary aim of this paper is to demonstrate a comprehensive KYC solution for car-sharing systems. More advanced algorithms for document validation will be explored beyond the scope of this research, but we plan to enhance this aspect in future work.

3.7 Extraction of Regions of Interest

The "identity number" and "driving license expiration date" are crucial data points for the car-sharing system. To enable customers to begin driving, it is essential to obtain and verify these values. The extraction process for these key data points is based on similar principles for both the identity number and the expiration date detection tasks.

The primary objective is to identify a unique feature, often a specific landmark on the document. Once this unique feature is located, we can determine the positions of the required data relative to it. This methodology allows us to effectively extract the necessary information, which is then passed on to the OCR step (Fig. 11).

In the context of Kazakhstani documents, unique landmarks may include a chip on the driving license or the facial images present on both documents. For chip detection, we employed the MaskRCNN model. Consistent with our previous methodology, we utilized 80% of the data for training and the remaining 20% for testing.

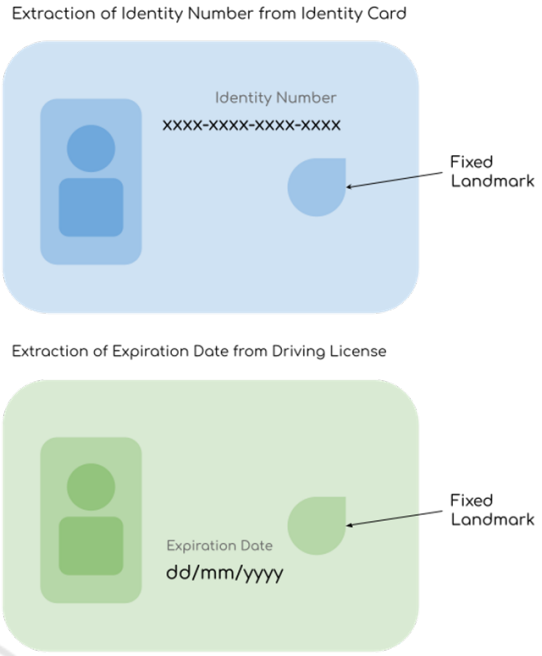

Figure 11: Extraction of ROIs using the fixed locations of document landmarks.

3.8 Optical Character Recognition (OCR) for Extracted ROIs

After the ROIs have been extracted and cropped, we can proceed with the OCR process. This process consists of three main components: 1) Text Detection, 2) Character Segmentation, and 3) Character Recognition. The initial step, text detection, has already been addressed in the previous section. We will now focus on character segmentation and recognition, utilizing open-source libraries like Tesseract, which automates these functions.

However, simply using Tesseract may not suffice, as the cropped text data can often include noise. Thus, it is essential to preprocess the data to eliminate any noise before applying Tesseract for OCR. The noise removal techniques typically include:

- Blurring the image (using methods such as average or Gaussian filtering)
- Histogram equalization
- A combination of morphological operations like erosion and dilation

Once the cropped text data has been adequately cleaned, we can apply Tesseract OCR. It is crucial to select the appropriate language during the character recognition phase. In the following subsection, we will integrate all these features to demonstrate how

to create a comprehensive system for automated customer onboarding.

3.9 Integration of ML with the end Application

All functionalities of the KYC system, provided as APIs, will be hosted on the ML server (see Fig. 12). Requests originating from the client side (mobile app) will be processed by the back-end server. This server will manage a queue of requests and forward them to the ML server for the appropriate API. Upon receiving a request from a specific customer, the ML server will process it and return the results to the back-end server. The back-end server will then relay these responses to the mobile app for display to the end users.

For future enhancements, we plan to implement a queue system to effectively manage customer load. This queue will be placed between the back-end server and the ML server.

4 ADDITIONAL TASKS

The KYC system has the potential to incorporate a variety of additional features. Among these are gender recognition and age estimation, which provide valuable insights into customer demographics. This information can later be leveraged for various purposes, such as marketing campaigns. The following subsection will detail the implementation of these additional features, focusing on age and gender prediction.

4.1 Demographic Attribute Prediction

Understanding the demographic attributes of customers is crucial for enhancing the KYC system. These attributes include location, gender, and age. While the customer's location can often be easily determined using GPS, identifying age and gender presents more complex challenges. In this research, we aim to tackle these tasks through Computer Vision and ML techniques.

Age prediction is treated as a regression task, where the output is a floating-point number representing an age within a specified range (e.g., 18 to 100 years). In contrast, gender classification is framed as a classification problem, where each input image must be categorized into one of a limited number of classes.

For both tasks, we utilize the Histogram of Oriented Gradients (HoG) as the feature extractor, producing feature vectors for each face (denoted as $f(i) = [f(i)_0, f(i)_1, \ldots, f(i)_{127}]$). These feature vectors serve as input for training the gender classifier and age predictor, which can be implemented using various methods such as Neural Networks or Support Vector Machines.

Let's denote the classifier as g. For the classification task, the model output is represented as $g(f(x(i)))$, where $x(i)$ denotes the i-th input image. This classification will yield a one-hot encoded output, such as $[0,0,\ldots,1,\ldots,0]$, where only one value is "1" and all others are "0." The index of the array with the value of "1" indicates the predicted class.

In the case of the regression task, $g(f(x(i)))$ will produce a floating-point number, as previously described. These prediction functions, $g(f(x(i)))$, can be integrated into the ML server as separate APIs. Whenever this demographic information is required, we can easily call these APIs from the back end.

5 FUTURE WORK

The current KYC solution presents several limitations. For instance, a customer could potentially take a photo of someone else and submit it as their selfie, thereby circumventing the ML system and gaining access to the platform with false information. This scenario poses significant risks, including vehicle theft, parts theft, and other criminal activities.

To address these vulnerabilities, we need to explore solutions that can effectively mitigate such risks. One promising approach is the implementation of "liveness detection." This feature aims to verify that the customer is a live individual and not merely presenting a printed photo as their selfie. The following subsection will delve into the specifics of the "liveness detection" feature.

5.1 Liveness Detection

Various methods exist for "liveness detection," which determine whether the individual on the client side is indeed a live person. We will utilize one of the most common techniques, which involves providing customers with a set of facial commands and monitoring their compliance. This method is referred to as "active liveness detection."

If the customer successfully follows the commands, we will conclude that they are a live individual and proceed with their onboarding. Conversely, if they fail to comply, their request will be rejected, and the KYC process will be sent for manual verification.

The list of facial commands may include:

- Open the mouth
- Close the eyes

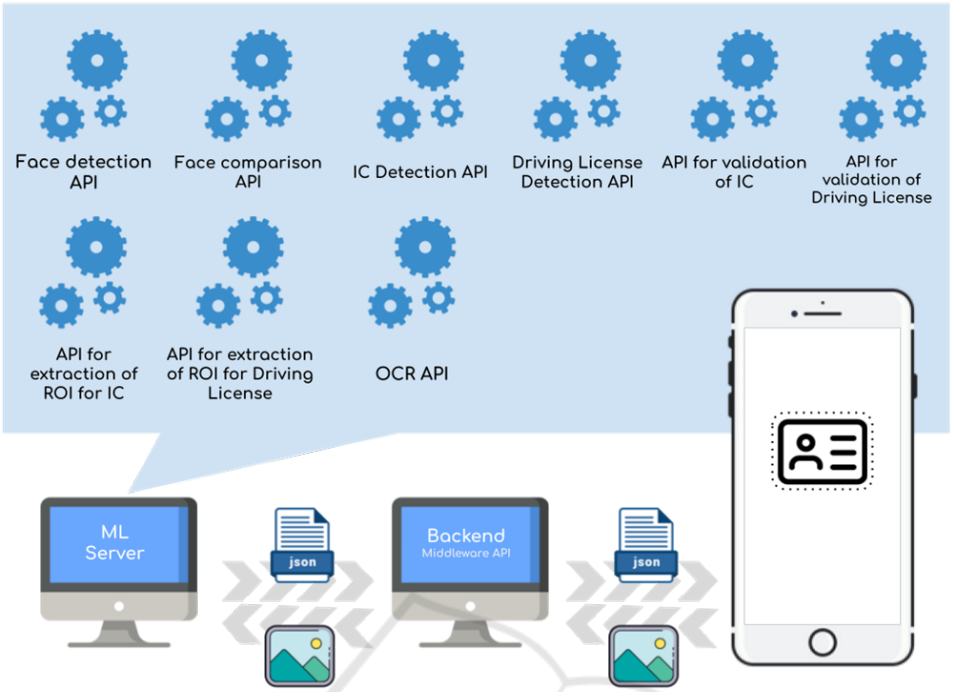

Figure 12: Comprehensive architecture of the ML-based KYC system.

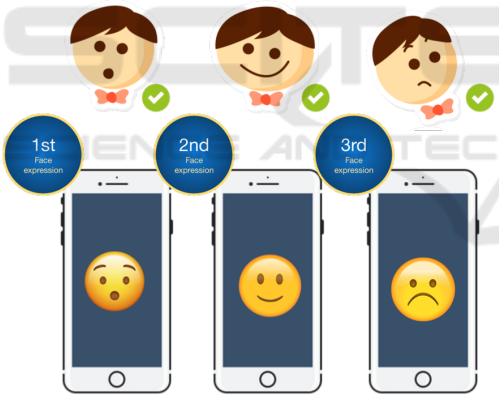

Figure 13: Illustration of the liveness detection process.

- Close the right/left eye
- Turn the head to the left/right
- Look up/down
- Etc.

We can randomly select three commands from the list above. Once a command is issued, we will classify the customer's actions. If the predicted action matches the intended command, we will proceed to the next command. If there are three correct matches out of three attempts, we will consider the person to be live; otherwise, the customer will not pass our test (Fig. 13).

5.2 Improving Photo Quality

Image quality can vary significantly based on the type of smartphone and its camera capabilities. Low-quality cameras and poor lighting conditions often result in subpar images, which can hinder the performance of the ML KYC system and create challenges for customers during onboarding. To address this, we can enhance image quality to achieve two main objectives: 1) improve text readability and 2) reduce noise.

To enhance text clarity, we can apply deblurring algorithms that focus on refining image quality, making input photos clearer and more legible. One effective approach for deblurring is to utilize Generative Adversarial Networks (GAN) (Lu et al., 2019). For reducing noise, we can implement various noise removal algorithms (Verma and Ali, 2013).

6 CONCLUSION

In this research, we have presented an architecture for developing a straightforward KYC solution suitable for various car-sharing use cases. Our proposed methods leverage Computer Vision and Machine Learning techniques, emphasizing the detection of regions of interest (ROIs) from images. Additionally, the ML server can be further enhanced by incorporating fea-

tures such as a gender classifier and an age predictor. To bolster the security of the KYC solution, we recommend implementing a "liveness detection" feature, which verifies whether the individual on the screen is indeed a live person.

REFERENCES

(2024). Cnn based face detector. Accessed: 2024-10-14.

(2024). Face recognition api. Accessed: 2024-10-14.

ACTICO (2023). Why machine learning brings up to 57% savings in the KYC process. Accessed: 2024-10-14.

Al, S. and Kumar, J. (2020). Fingerprint scanning technology in modern identity verification systems. *International Journal of Computer Applications*, 28(3):65–78.

Bhardwaj, R. and Sharma, P. (2021). Anomaly detection algorithms in anti-money laundering systems. *Journal of Artificial Intelligence Research*, 34(4):45–60.

Cerovsky, Z. and Mindl, P. (2008). Hybrid electric cars, combustion engine driven cars and their impact on environment. In *2008 International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, pages 739–743. IEEE.

Chang-Yeon, J. (2008). Face detection using lbp features. *Final Project Report*, 77:1–4.

Charoenwong, B. (2023). The one reason why AI/ML for AML/KYC has failed (so far). Accessed: 2024-10-14.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee.

Darapaneni, N., Evoori, A. K., Vemuri, V. B., Arichandrapandian, T., Karthikeyan, G., Paduri, A. R., Babu, D., and Madhavan, J. (2020). Automatic face detection and recognition for attendance maintenance. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pages 236–241. IEEE.

Do, T.-L., Tran, M.-K., Nguyen, H. H., and Tran, M.-T. (2021). Potential threat of face swapping to eKYC with face registration and augmented solution with deepfake detection. In *Future Data and Security Engineering: 8th International Conference, FDSE 2021, Virtual Event, November 24–26, 2021, Proceedings 8*, pages 293–307. Springer.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Jain, A. K. and Gupta, S. B. (2017). Facial recognition: Advances and applications in identity verification. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(2):115–130.

Kagawa, S., Hubacek, K., Nansai, K., Kataoka, M., Managi, S., Suh, S., and Kudoh, Y. (2013). Better cars or older cars?: assessing co2 emission reduction potential of passenger vehicle replacement programs. *Global Environmental Change*, 23(6):1807–1818.

Leydesdorff, L. (2008). On the normalization and visualization of author co-citation data: Salton's cosine versus the jaccard index. *Journal of the American Society for Information Science and Technology*, 59(1):77–85.

Liu, X. and Zhang, Y. (2019). Blockchain technology for secure kyc in financial systems. *Journal of Blockchain Research*, 5(2):45–59.

Lu, B., Chen, J.-C., and Chellappa, R. (2019). Uid-gan: Unsupervised image deblurring via disentangled representations. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(1):26–39.

Miller, M. and Smith, J. (2019). Biometric authentication systems in financial applications. *Journal of Financial Technology*, 12(3):89–104.

Mita, T., Kaneko, T., and Hori, O. (2005). Joint haar-like features for face detection. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1619–1626. IEEE.

Pic, M., Mahfoudi, G., and Trabelsi, A. (2019). Remote KYC: Attacks and counter-measures. In *2019 European Intelligence and Security Informatics Conference (EISIC)*, pages 126–129. IEEE.

Recognition, F. Face comparison distance calculation. Accessed: 2024-10-14.

Stefanovic, S. (2023). Face detection algorithms comparison. Accessed: 2024-10-14.

Sun, X., Wu, P., and Hoi, S. C. (2018). Face detection using deep learning: An improved faster rcnn approach. *Neurocomputing*, 299:42–50.

Technologies, S. (2023). How we built an intelligent automation solution for KYC validation. Accessed: 2024-10-14.

Verma, R. and Ali, J. (2013). A comparative study of various types of image noise and efficient noise removal techniques. *International Journal of advanced research in computer science and software engineering*, 3(10).

Yoon, J. and Kim, R. (2020). Pattern recognition in financial fraud detection systems. *IEEE Transactions on Computational Intelligence*, 9(5):98–112.

Zohdy, M. and Thomas, L. (2018). Blockchain technology for secure customer data sharing in kyc systems. *Blockchain Technology Review*, 2(1):23–37.