BLOCKCHAIN BASED E-VOTING SYSTEM

FINAL SEMINAR

SUPERVISORS

Dr. Dina Elsayad

TA. Manar Sultan

TEAM MEMBERS

Suhail Mahmoud

Heba Shaaban

Abdelrahman Hamdi

Rana Ahmed

Abdelrahman Osama

Habiba A. Abdulkader

AGENDA

01 Problem Definition

02 Motivation

03 Objective

04 System Architecture

05 Data Creation

06 Face Verification

07 ID Data Extraction

08 Fake ID Detection

09 Blockchain

10 Frontend

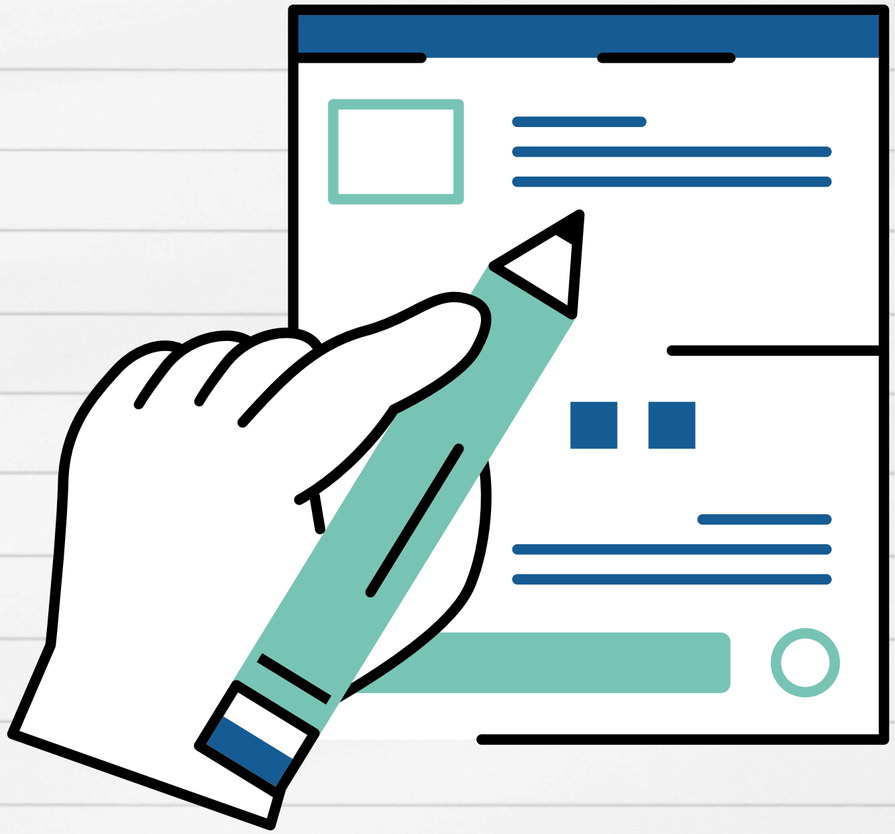
11 Integration

12 Conclusion

13 Future Work

01

PROBLEM DEFINITION

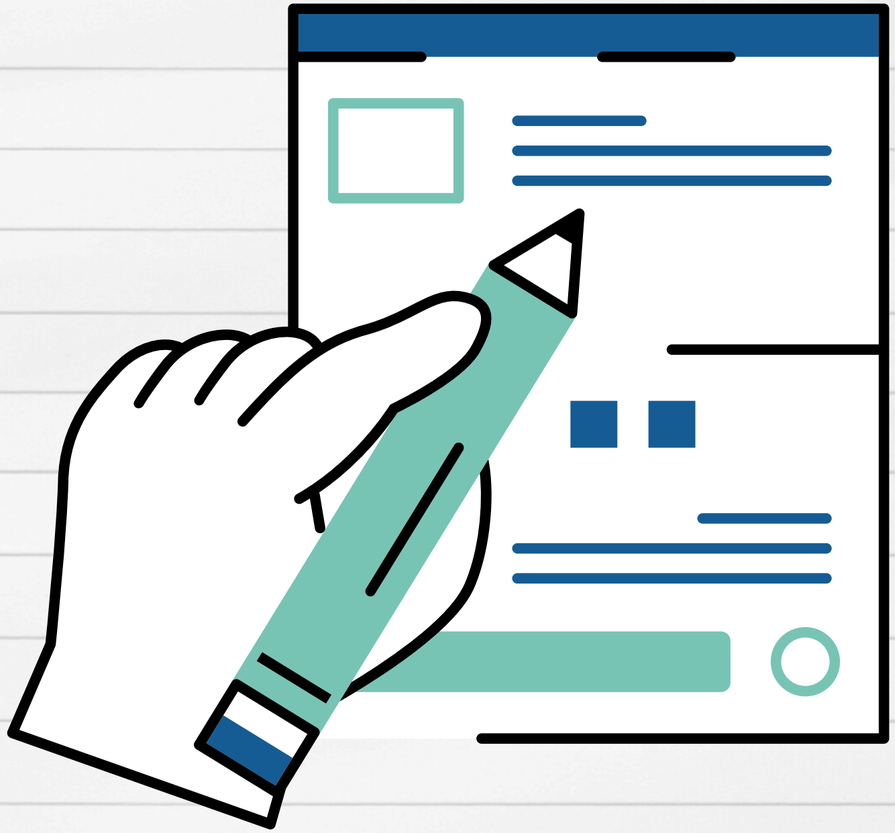


PROBLEM DEFINITION

Current voting systems face various security and transparency issues, including fraud, manipulation, and lack of trust from the public. Traditional systems make it challenging to securely identify voters while keeping their identities private. These vulnerabilities highlight the need for a reliable, decentralized solution to ensure vote integrity, transparency, and user privacy.

02

MOTIVATION

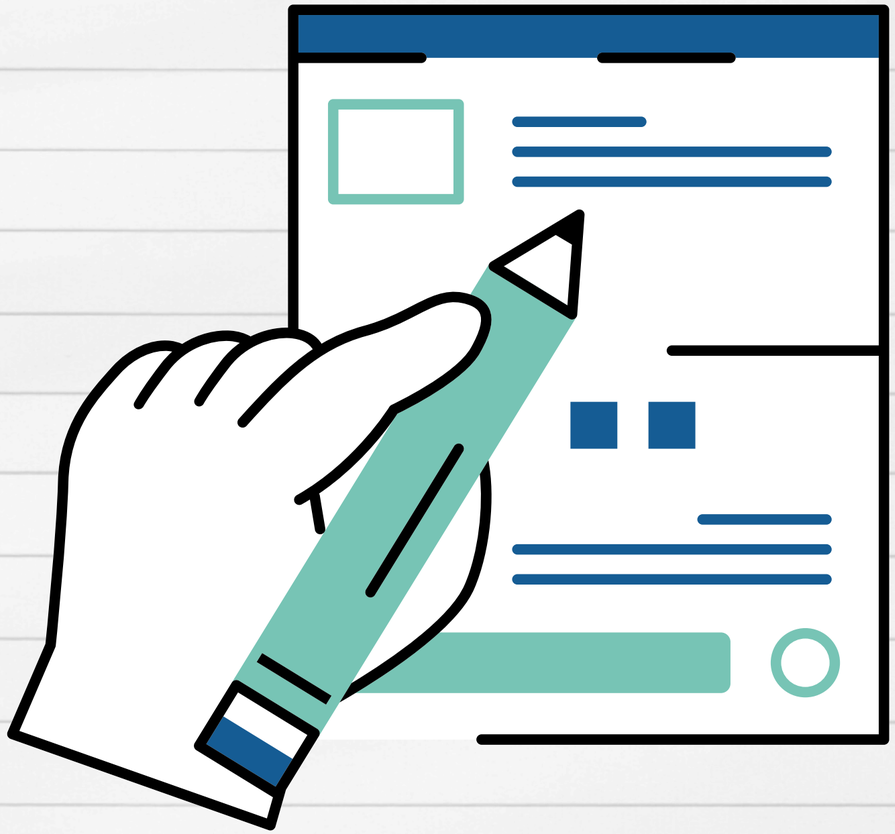


MOTIVATION

The main motivation is to build a secure and transparent e-voting platform that leverages blockchain's decentralized structure to provide an immutable record of votes. Additionally, it utilizes AI technology for user authentication, ensuring that only verified users can participate and that each vote remains anonymous and unaltered.

03

OBJECTIVES

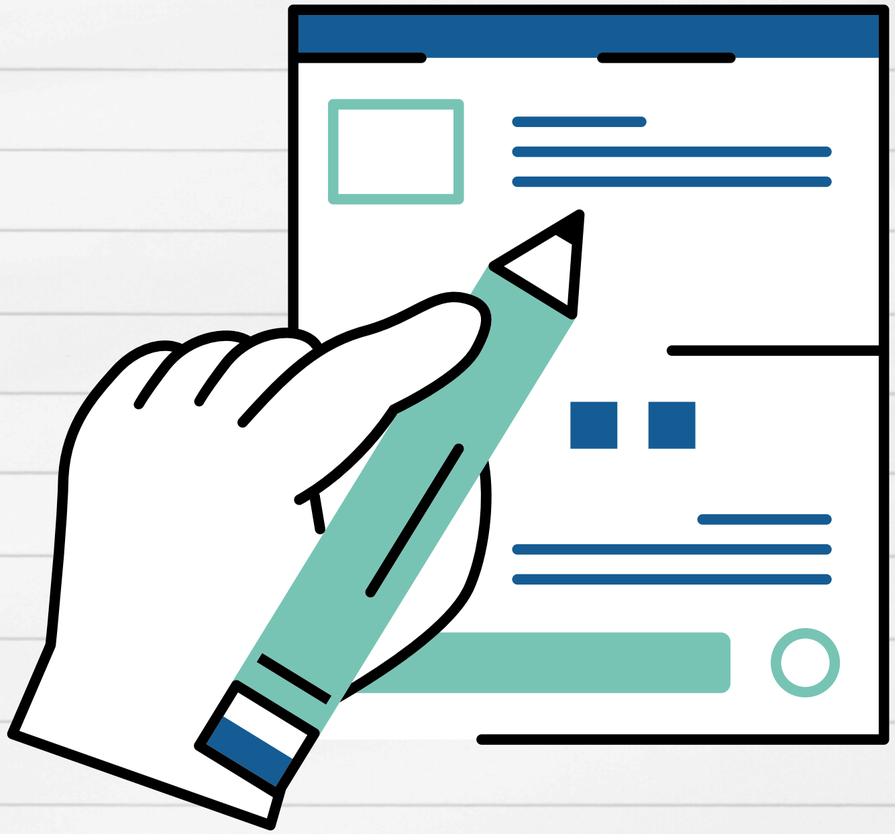


OBJECTIVE

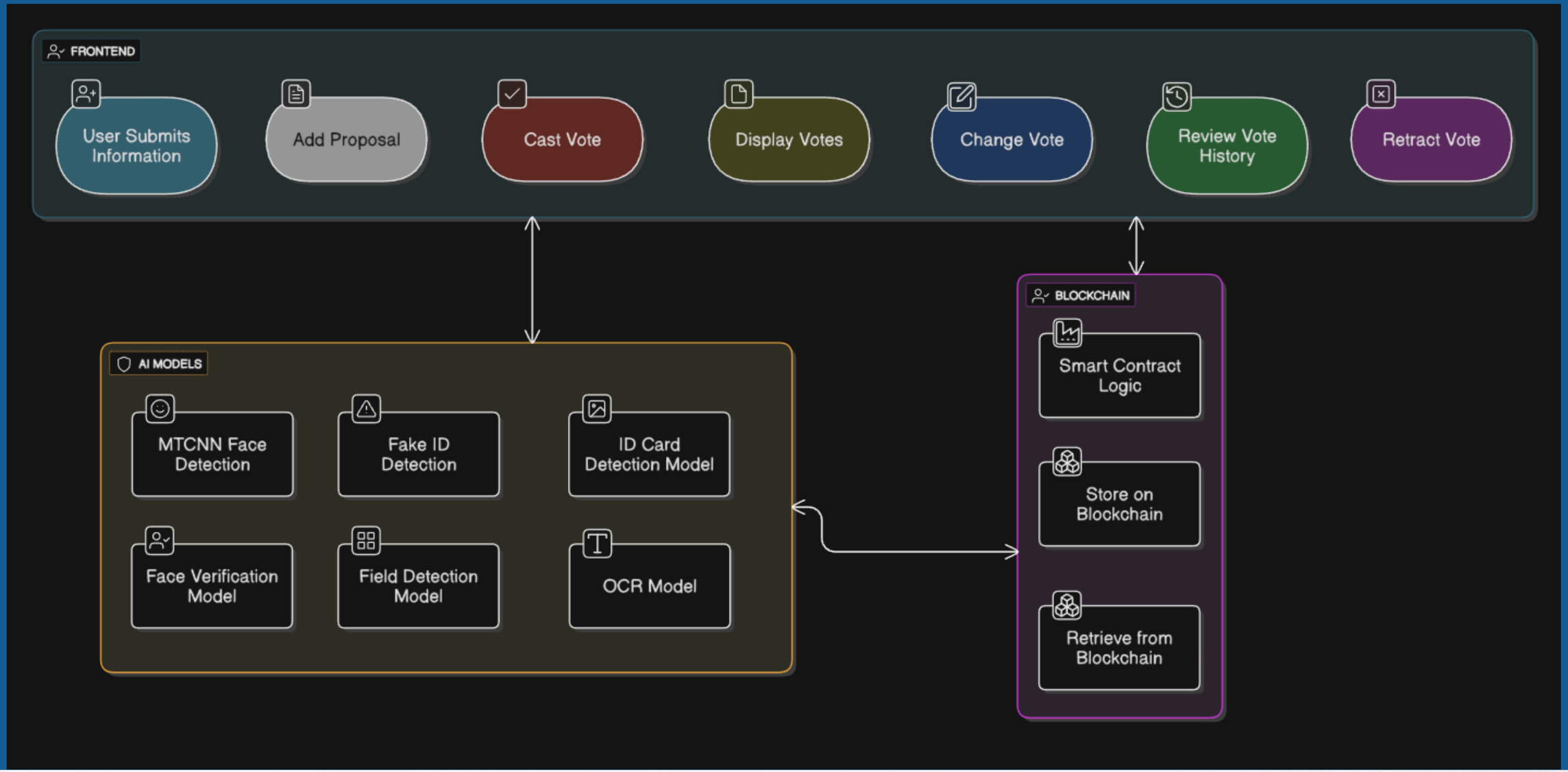
- The main goal of this project is to develop an online voting system with blockchain as its core foundation and AI-powered identity verification. This system will :
  - Ensure privacy and trust in voting processes, benefiting governments, organizations, and private entities seeking secure voting solutions.
  - Enable real-time, auditable, and transparent voting records.

04

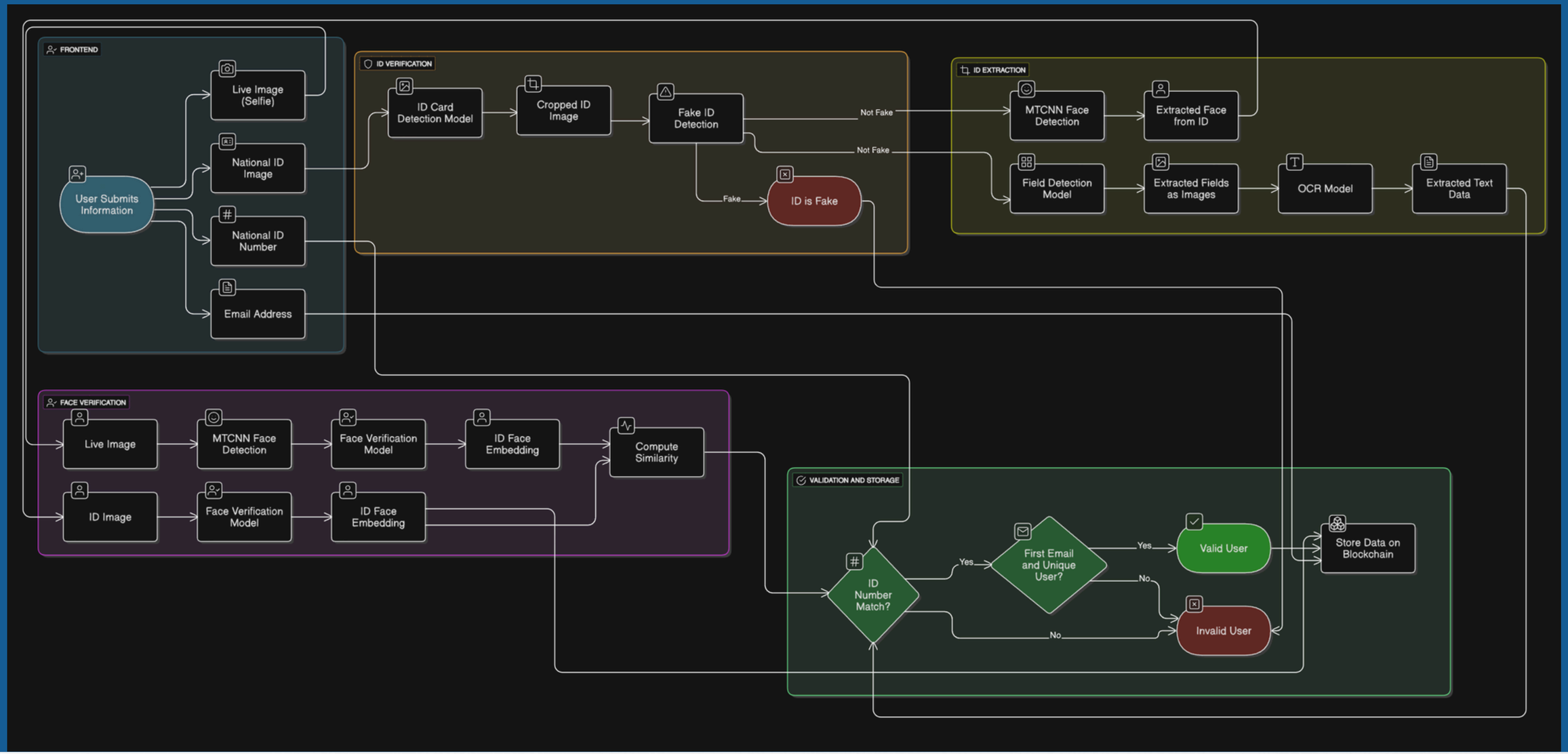
SYSTEM ARCHITECTURE



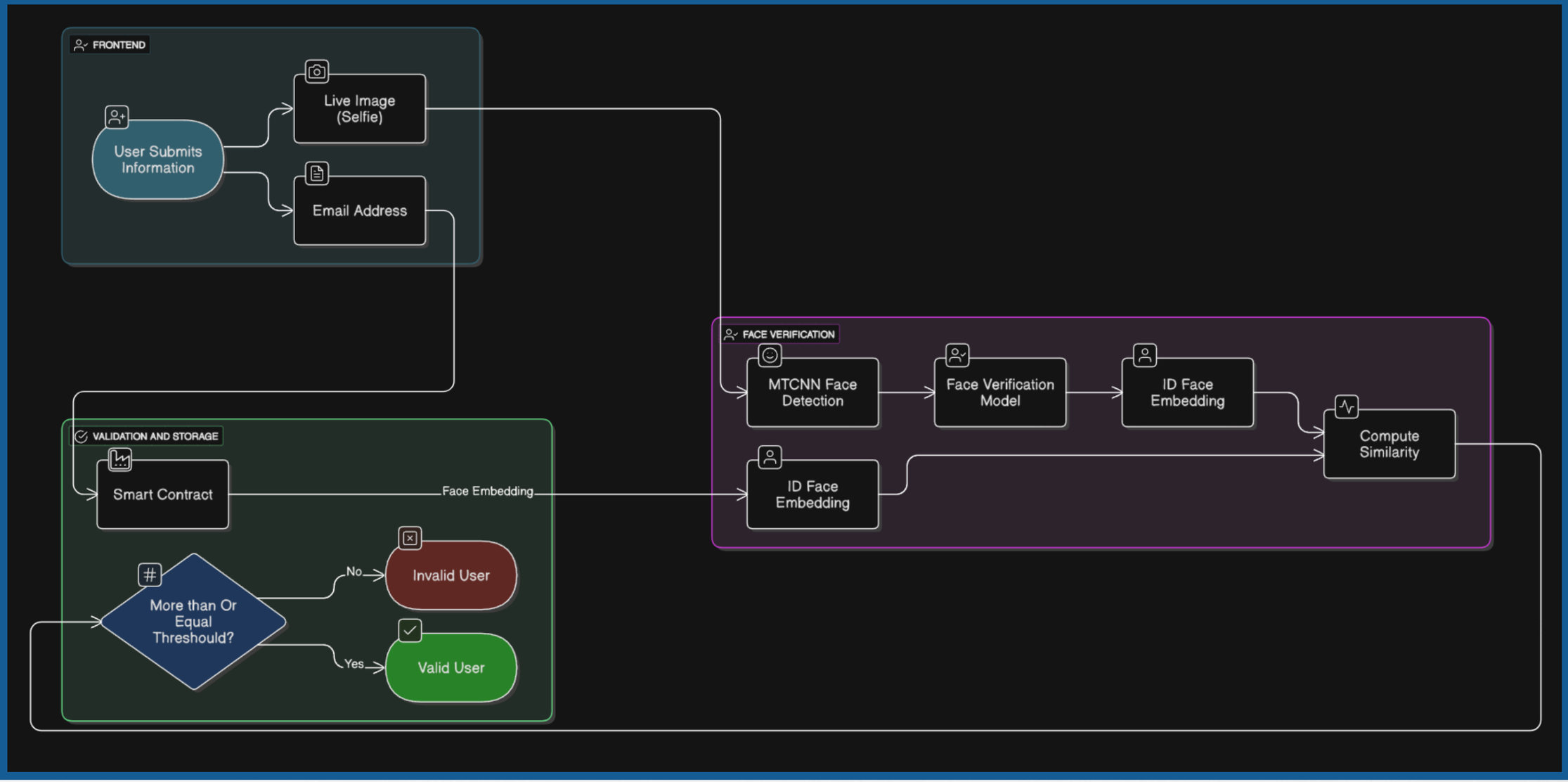
ENTIRE SYSTEM ARCHITECTURE



REGISTRATION PHASE FLOW

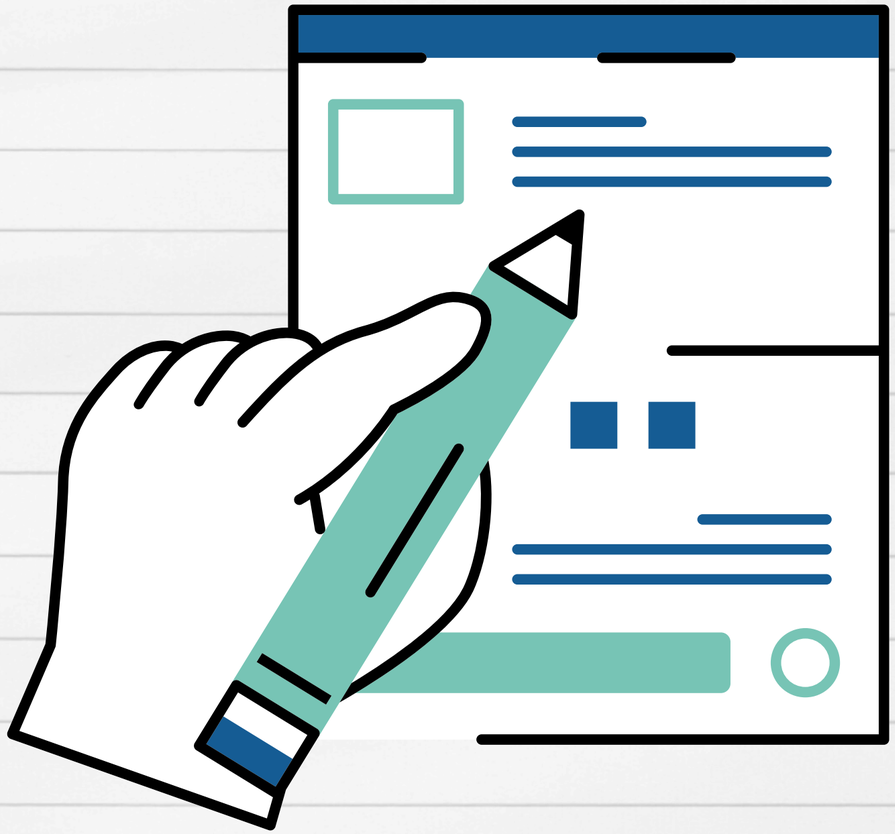


LOGIN PHASE FLOW



05

DATA CREATION



DATA CREATION

We needed to collect and create Egyptian national ID Dataset.

- **Security:** Verifying the user’s national id, we can prevent impersonation and unauthorized access.
- **Integration:** AI verification integrates seamlessly with the blockchain to reduce administrative overhead.

DATA CREATION

- Collecting all the data we can find, filtering to only keep **Unique** data
- Websites (Roboflow, Facebook, google, github.....).
- Number of Data over **32K** image.
- After filtering, only **8,649** image (probably).

DATA CREATION

Filtering

- **manually**.
- Using some **regex** technique for names of images.
- Using **CNN** Network like Simple CNN Network MobileNetV2 with **manual** review.

4222.jpg

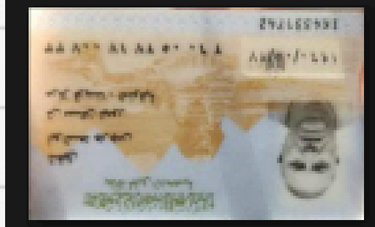
4223.jpg

4224.jpg

4225.jpg

4226.jpg

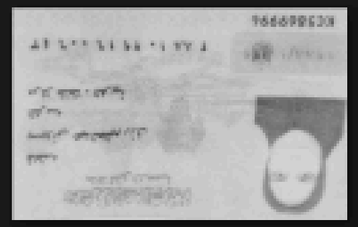
4227.jpg



4233.jpg

4234.jpg

4235.jpg



4236.jpg

4237.jpg

4238.jpg

4244.jpg

4245.jpg

4246.jpg

4247.jpg

4248.jpg

4249.jpg

DATA CREATION

Unique Faces From

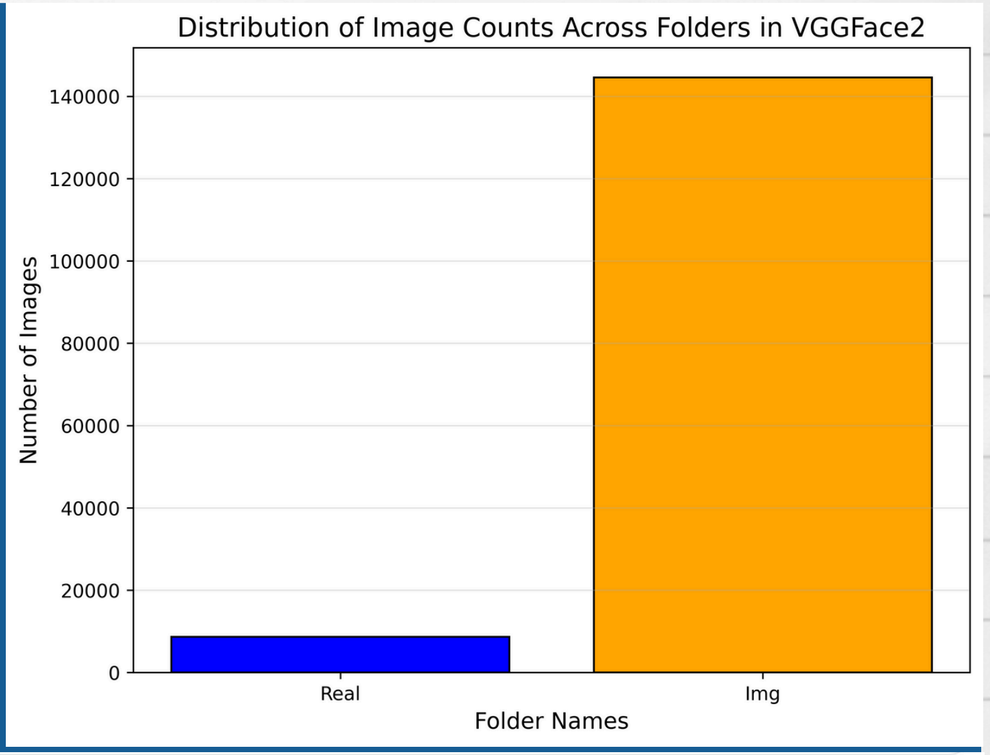
- VggFace2 (8631 identity).
- CelebA (10177 identity).
- Synthetic Faces High Quality (SFHQ) part 4 (125K identity).

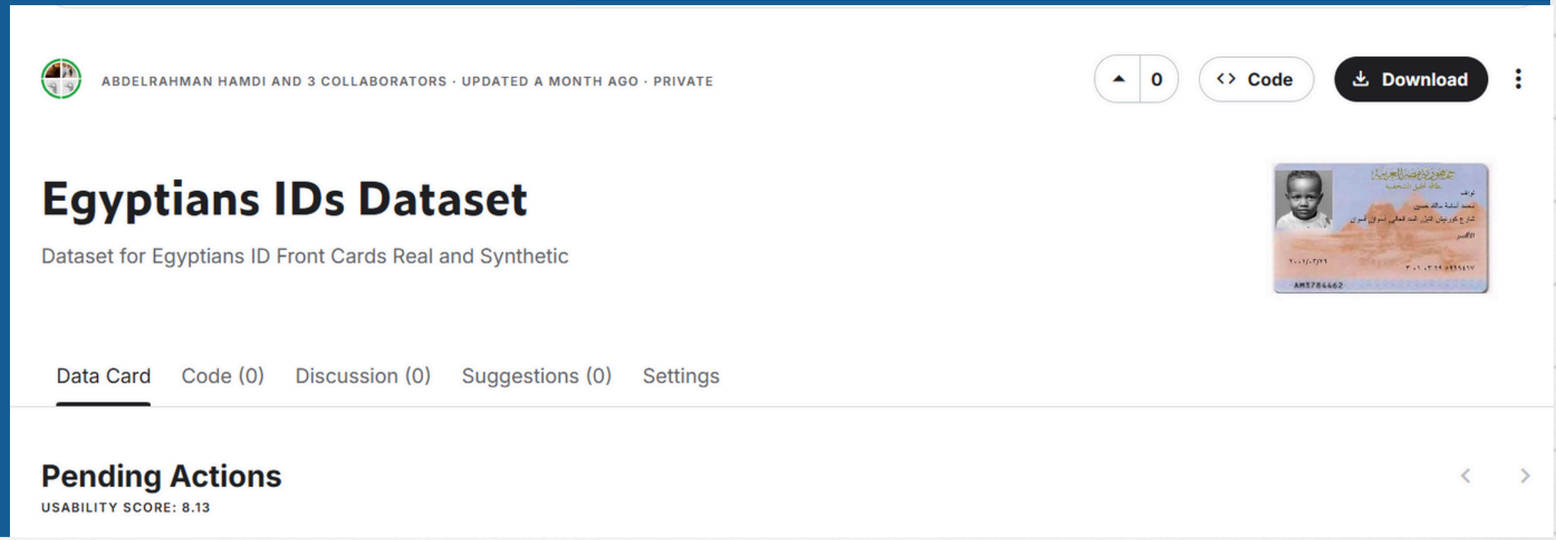
DATA CREATION

DATA CREATION

With **8649** real & over **144K** Img, Now, we had over **153K** Egyptian ID Card.

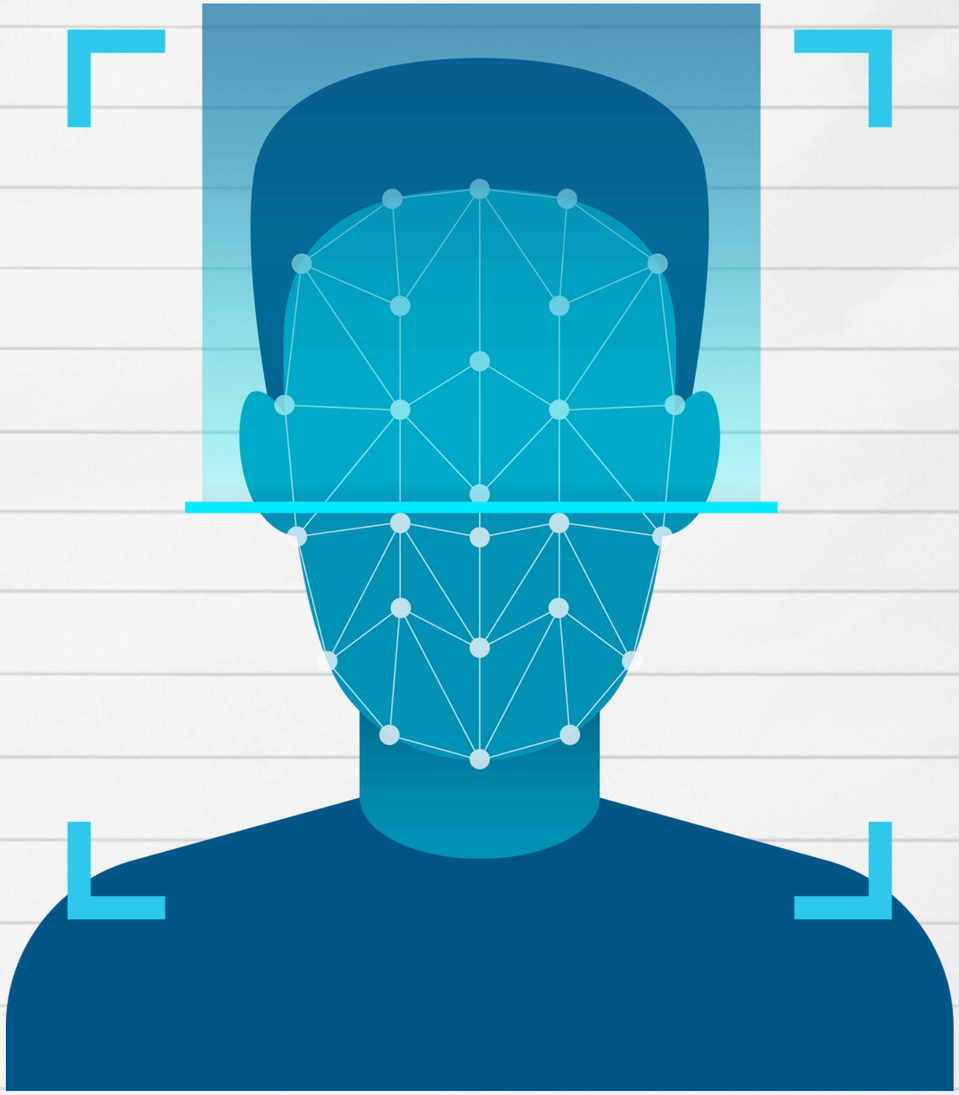
it is now on Kaggle.





06

FACE VERIFICATION



FACE VERIFICATION

- **WHAT IS THAT?**
  - AI-based facial recognition module will ensure voter authentication.

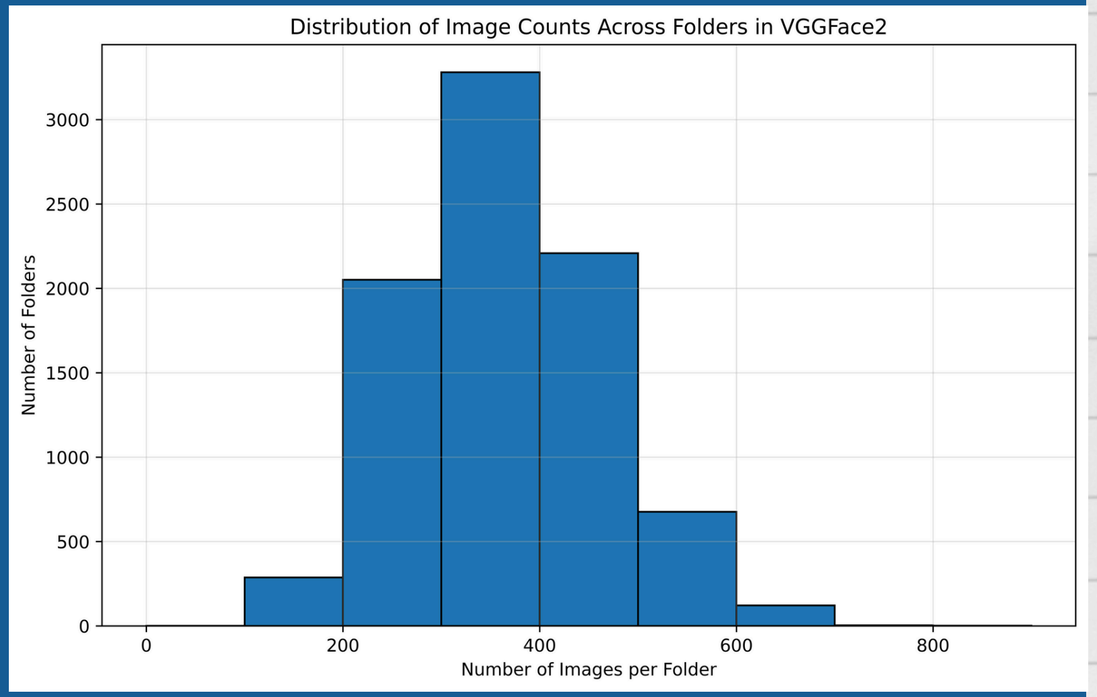
Dataset: VGGFace2

- **Overview**
  - Includes over 3.3 million images.
  - Contains over 9,000 identities.
  - Diversity in terms of pose, ethnicity, and lightning conditions.

- **Features**
  - **Pose Variation:** Includes faces captured at various angles.
  - **Expression Diversity:** Includes images with a wide range of facial expressions.
  - **Backgrounds:** Includes images with varied backgrounds to improve generalization.

Analysis

- Number of Identities: 8631
- Number of Images: 3.14M
- Min image per folder: 87
- Max image per folder: 843
- Avg images per folder: 364.02



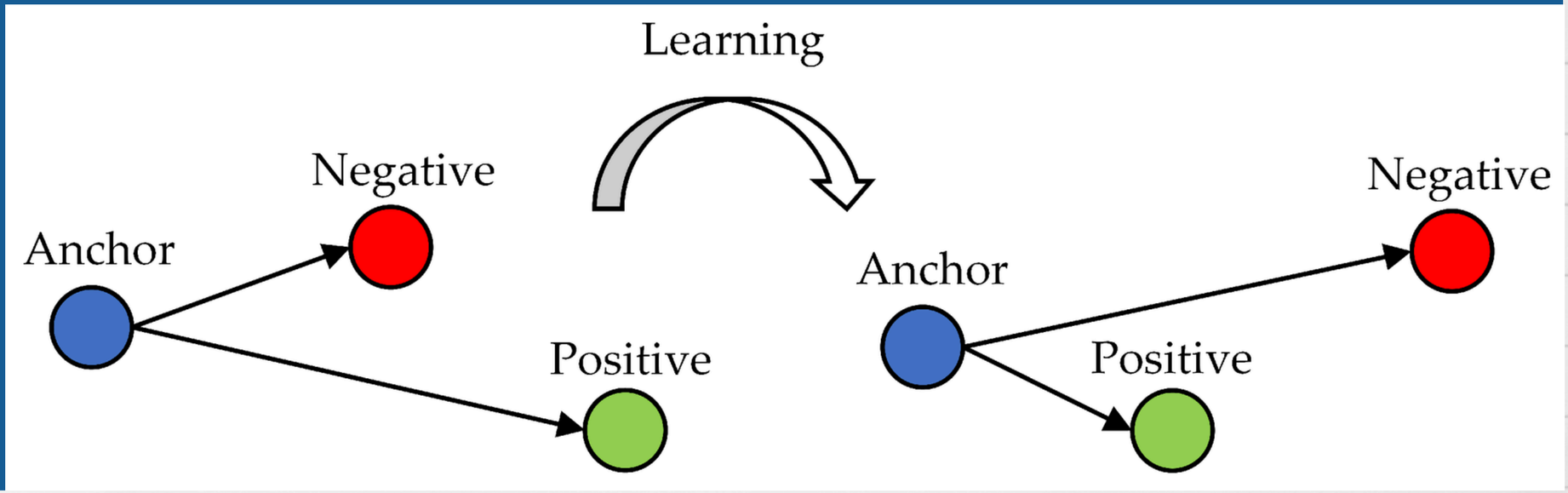
Preprocessing

- Detect Face (MTCNN)
- Resize (160 x 160)
- Augmentation (Horizontal Flip, Rotation, Color)

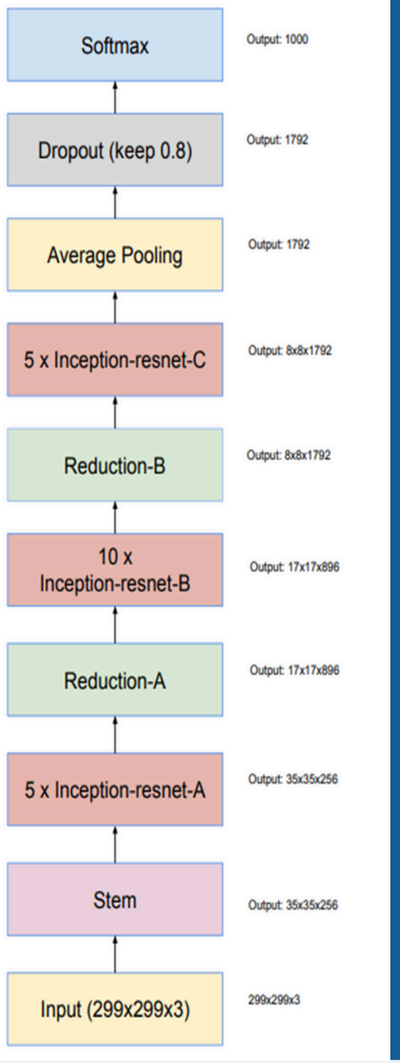
Triplets

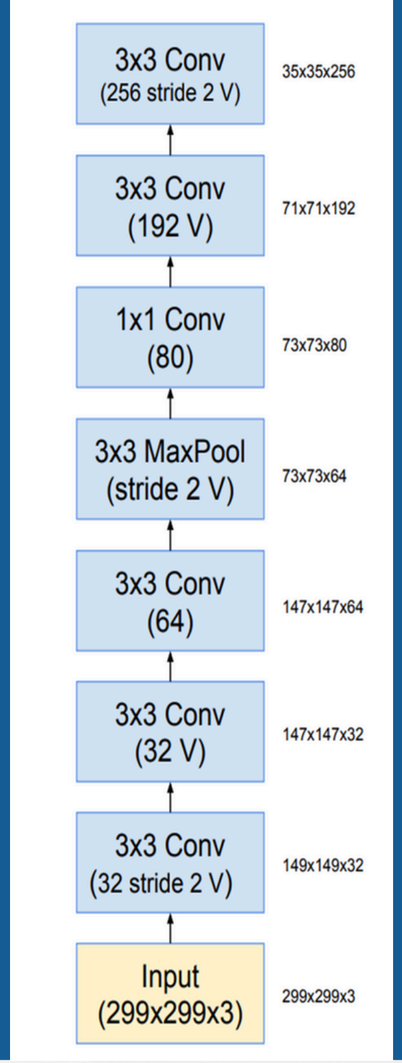
Anchor , Positive from the Same Class and Negative from different Class

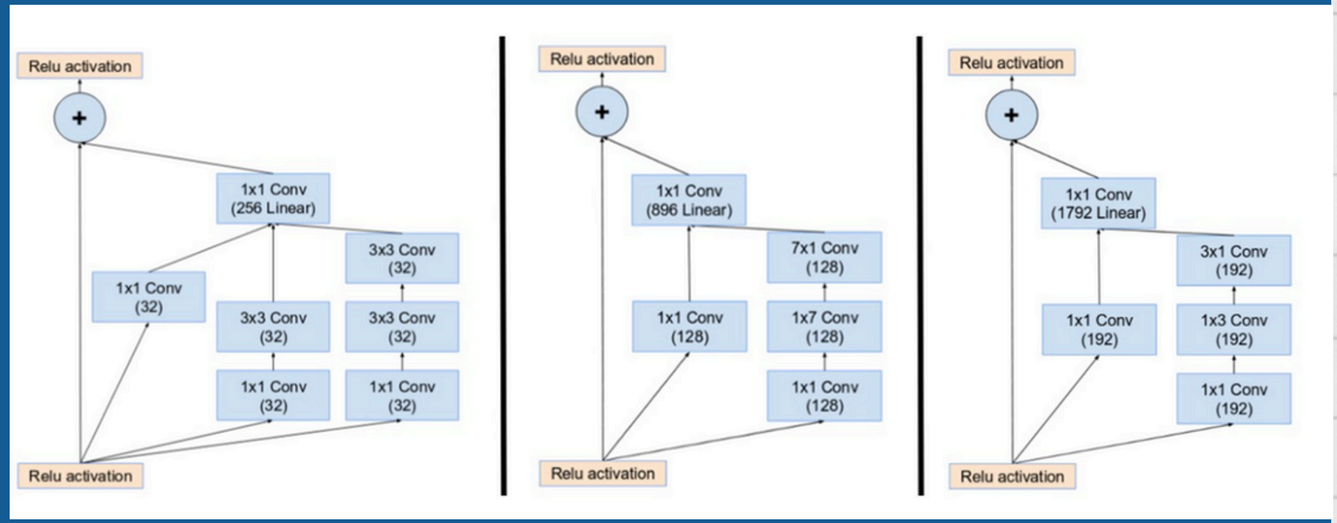
Model Learn to Increase distance between anchor and negative.
Decrease distance between anchor and positive.

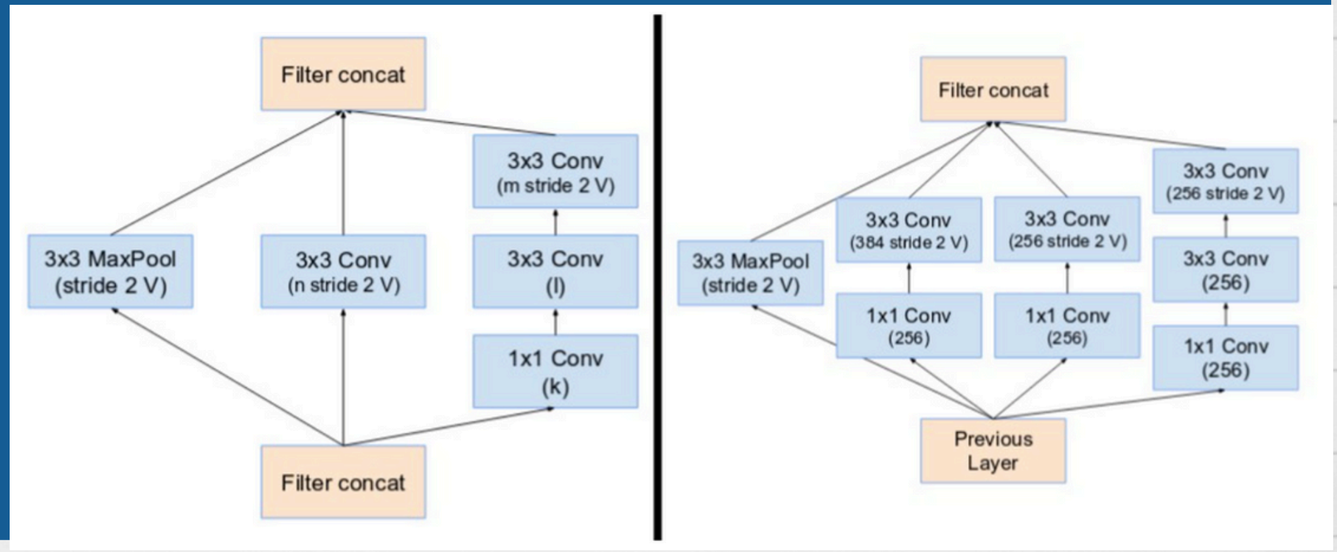


Model: InceptionResnet-V1



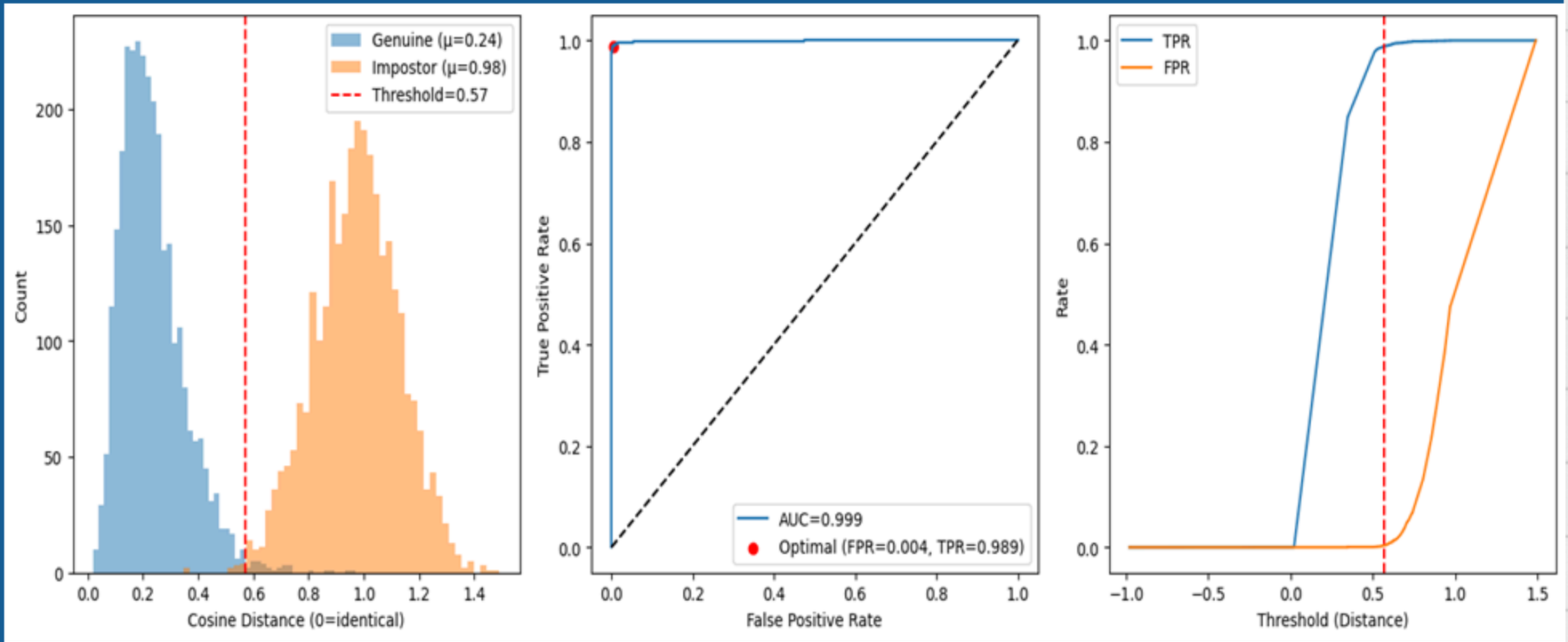






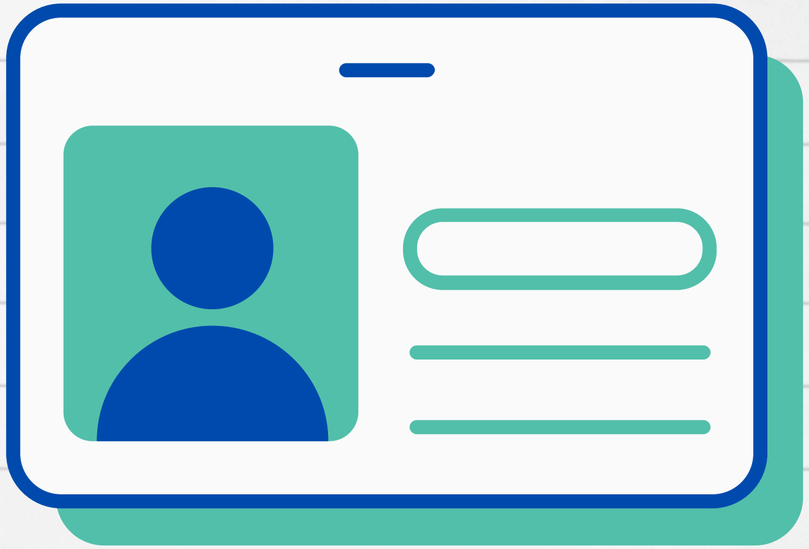
Result

Accuracy: 99.28%
Precision: 99.63%
Recall (TPR): 98.93%
F1-score: 99.28%
FPR: 0.37%
EER: 0.87%
Optimal Threshold: 0.5698
AUC: 0.999
Average Precision: 0.999

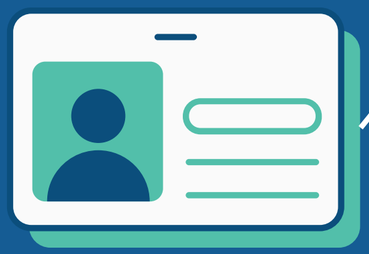


08

ID VERFICATION



ID VERIFICATION



ID Detection model

Fields Detection model

OCR model

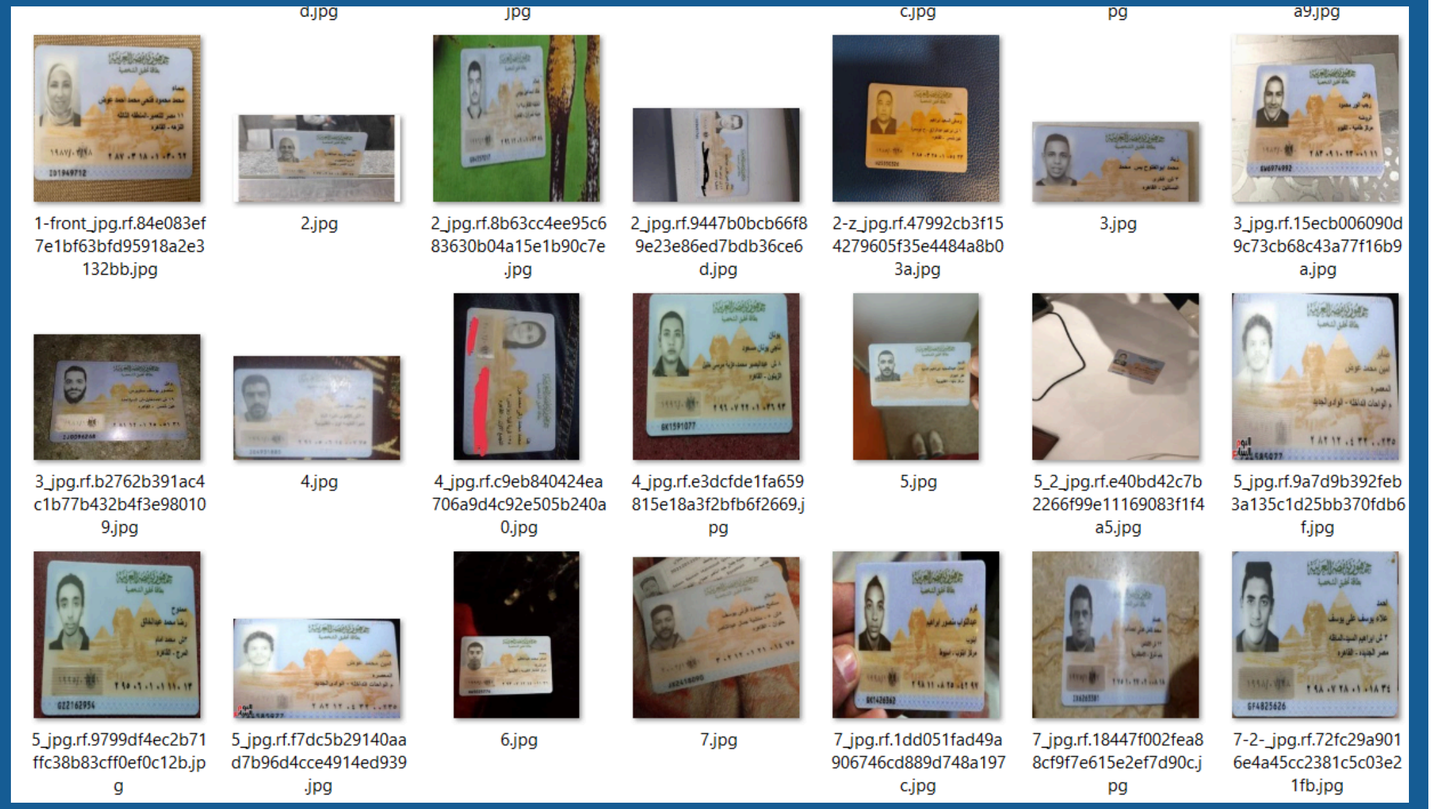
DataSet:

- **Dataset Name:** **Egyptian-ID-Dataset we have been collected from different resources**

Usage Breakdown:

  - **ID Detection:** 4200 images to detect ID cards within larger images
  - **(Fields Detection):** 150 images to detect fields like Name, ID Number, Birthdate, etc.

DataSet:


d.jpg
jpg
c.jpg
pg
a9.jpg
1-front_jpg.rf.84e083ef7e1bf63bfd95918a2e3132bb.jpg
2.jpg
2_jpg.rf.8b63cc4ee95c683630b04a15e1b90c7e.jpg
2_jpg.rf.9447b0bcb66f89e23e86ed7bdb36ce6d.jpg
2-z_jpg.rf.47992cb3f154279605f35e4484a8b03a.jpg
3.jpg
3_jpg.rf.15ecb006090d9c73cb68c43a77f16b9a.jpg
3_jpg.rf.b2762b391ac4c1b77b432b4f3e980109.jpg
4.jpg
4_jpg.rf.c9eb840424ea706a9d4c92e505b240a0.jpg
4_jpg.rf.e3dcfde1fa659815e18a3f2bfb6f2669.jpg
5.jpg
5_2_jpg.rf.e40bd42c7b2266f99e11169083f1f4a5.jpg
5_jpg.rf.9a7d9b392feb3a135c1d25bb370fdb6f.jpg
5_jpg.rf.9799df4ec2b71ffc38b83cff0ef0c12b.jpg
5_jpg.rf.f7dc5b29140aad7b96d4cce4914ed939.jpg
6.jpg
7.jpg
7_jpg.rf.1dd051fad49a906746cd889d748a197c.jpg
7_jpg.rf.18447f002fea88cf9f7e615e2ef7d90c.jpg
7-2-_jpg.rf.72fc29a9016e4a45cc2381c5c03e21fb.jpg


89a7fad913ddb69d9688.jpg
04ec2fee1444ff0a1928.jpg
18_jpg.rf.f012a7bfbc65c03f1c02851c310dae70.jpg
18-frontImage_1_jpeg.rf.4377d00d1d28626bb49debe87720c800.jpg
18-z_jpg.rf.facfabf0cbe923016e6594173242b55a.jpg
19.jpg
20_jpg.rf.3f8b564a9cd224f3e531efb12ec38d51.jpg
20_jpg.rf.8ef7d4bc76507a21dbe1fdfbdb9a894d.jpg
20_jpg.rf.90ff7db7de53cef63389f95b6bc34116.jpg
20-2-_jpg.rf.734eda47fcd6096b5f8f7620364638e5.jpg
22_jpg.rf.f0ed6d9e16b32ca15c241f89a8d9bf0a.jpg
22-z_jpg.rf.8118cb479d67634ec21473f1d524576b.jpg
23.jpg
23_jpg.rf.f8531913ee4aacf204bb85556f41c058.jpg

Preprocessing & Data Augmentation

Salt and paper noise , Brightness adjustment , Gaussian blur, Rotating

Dataset Overview & Usage

01

ID Detetction
DataSet
(4200 image)

02

Fileds Detection
DataSet
(150 image)

Datasets Pipeline Using Roboflow

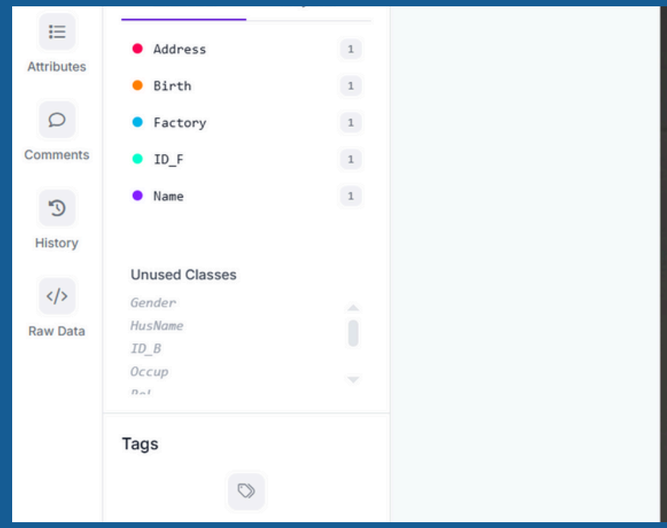
1. Labeling

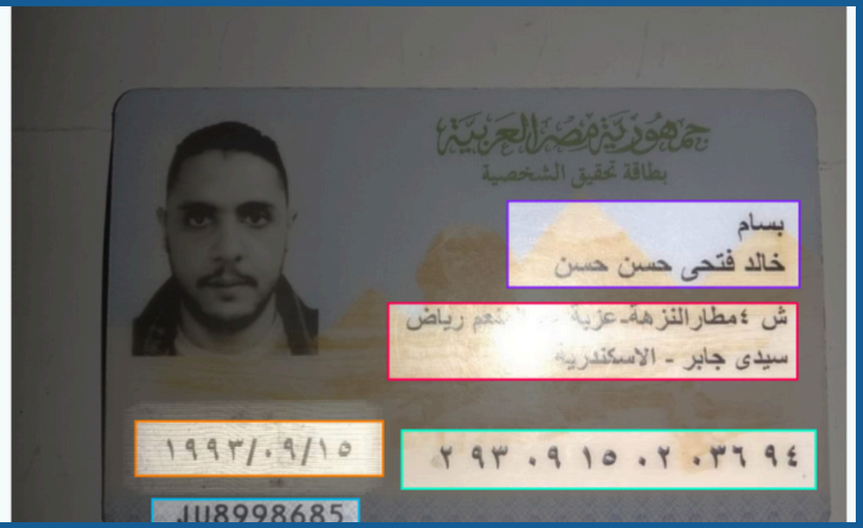
2. Splitting

3. dataSet.yaml (Exporting)

Roboflow made it easier to manage data and avoid bias during training

Labeling using Roboflow Tools



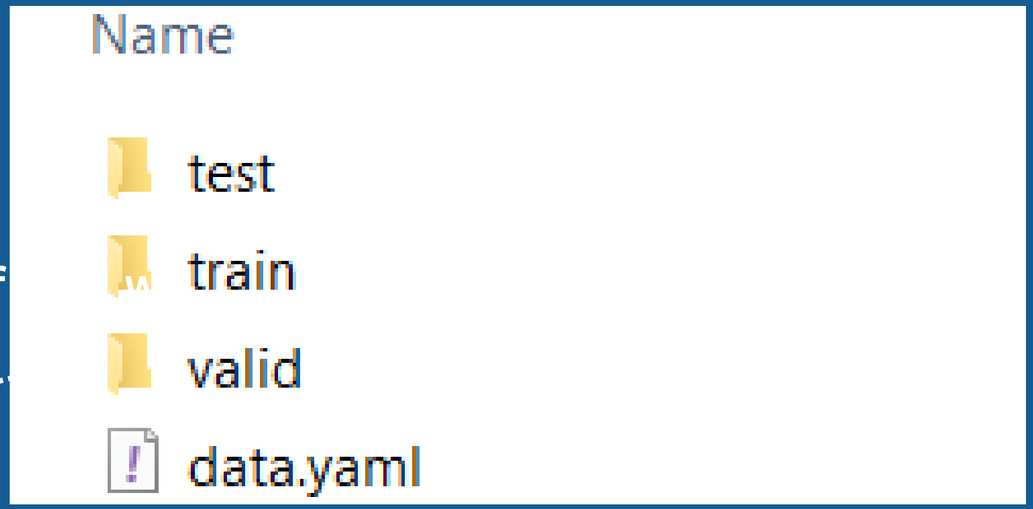



```
File  Edit  Format  View  Help
7 0.735830078125 0.3846398891966758 0.32576171875 0.15905817174515235
0 0.71365234375 0.5599861495844876 0.37291015625 0.15361495844875345
6 0.678369140625 0.768393351800554 0.4741796875 0.0915927977839335
1 0.24103515625 0.7428670360110804 0.246728515625 0.10781163434903047
2 0.238525390625 0.8764542936288088 0.22384765625 0.07819944598337951
```

class_id ,x_center, y_center, width ,height

🗂️ splitting Dataset using YOLOv8 format

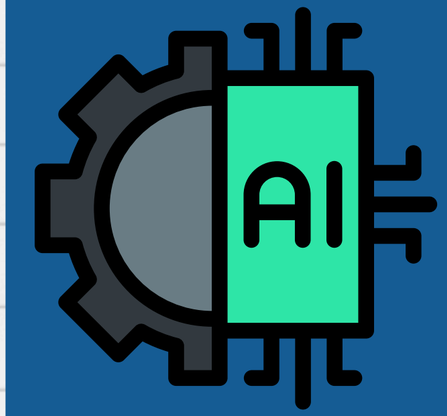
- DataSet Splits into:
  - 70% Training
  - 20% Validation
  - 10% Testing
- Annotated via Roboflow



After collecting and labeling the dataset in Robof
downloaded the YOLOv8-compatible format”

Data Set Strucure :

📂 train/
├── images/
└── labels/ ← Each .txt file contains bounding boxes for fields in the image
📂 valid/
📂 test/
📄 data.yaml ← Defines class names and paths

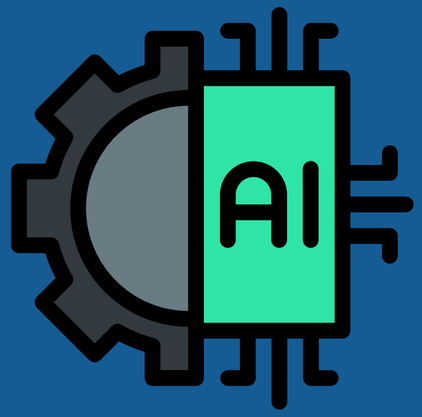


Let’s fine-tune our models

Model Selection

Why we Choose Yolov8 ?

We Used YOLOv8n.pt from Ultralytics
Because it’s Lightweight and fast
good for prototyping

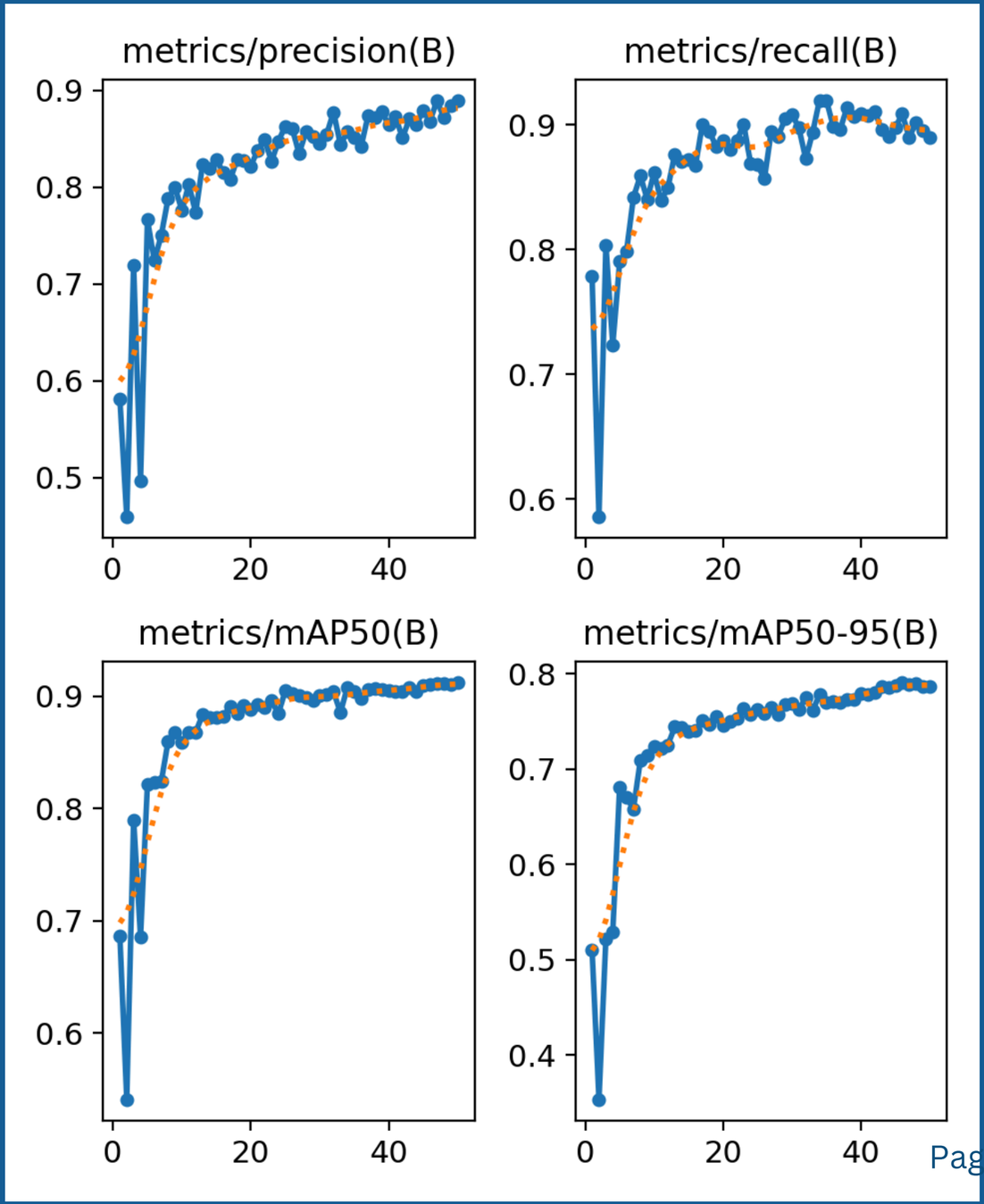


Model 1: Detect ID Card inside image

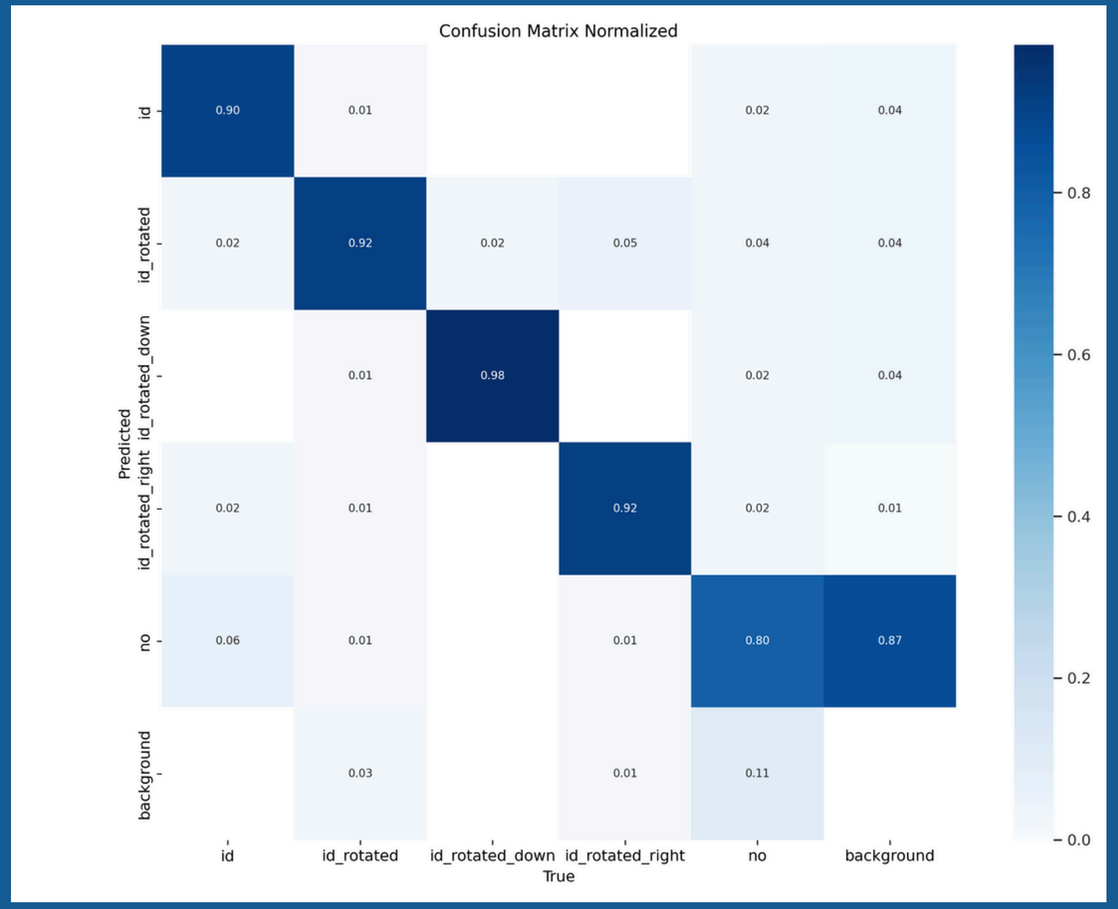
Model Performance

Started at 0.58 (epoch 1), reached 0.89 by epoch 50

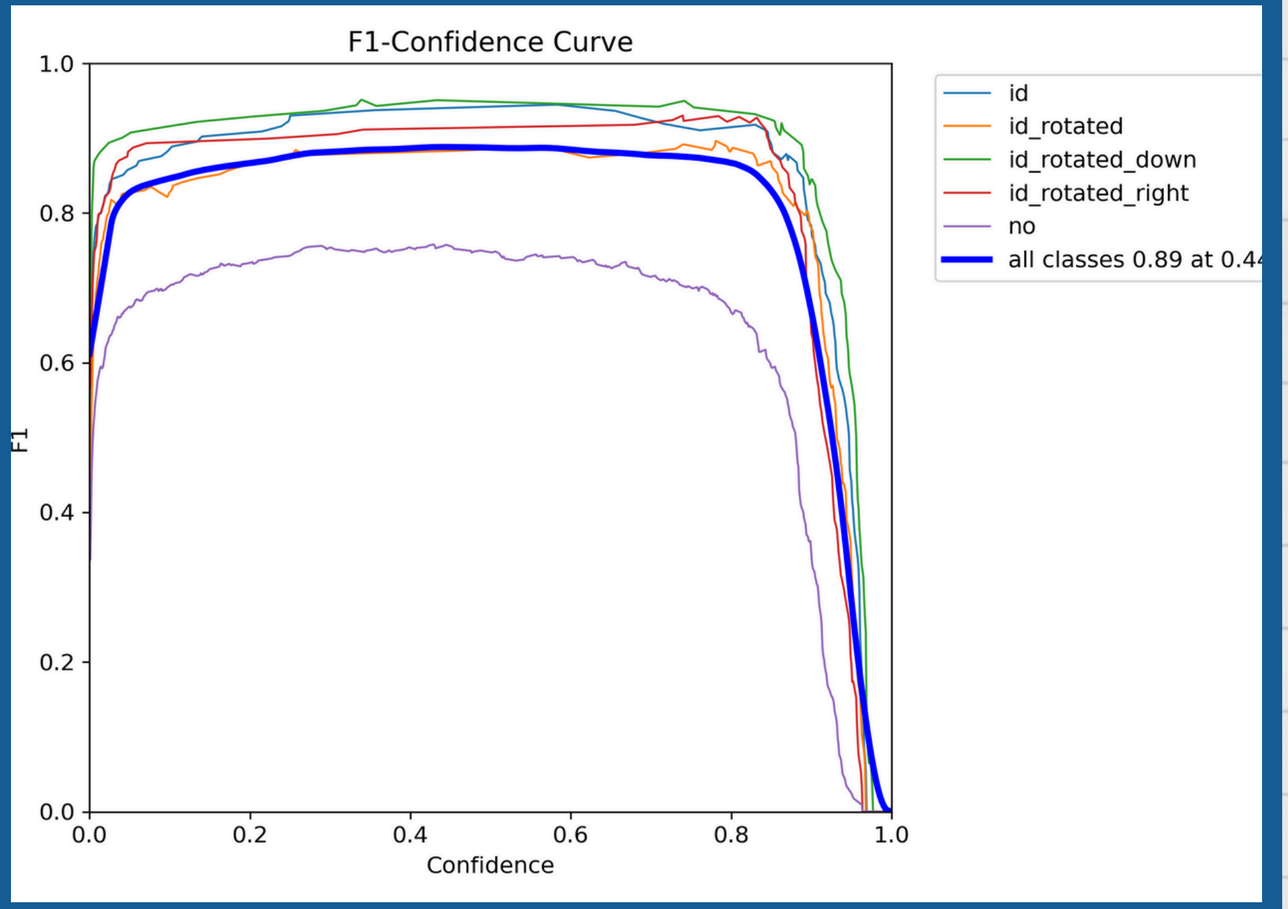
- **Precision (B): 0.890**
- **Recall (B): 0.890**
- **mAP50 (B): 0.912 (IOU)**



Confusion Matrix

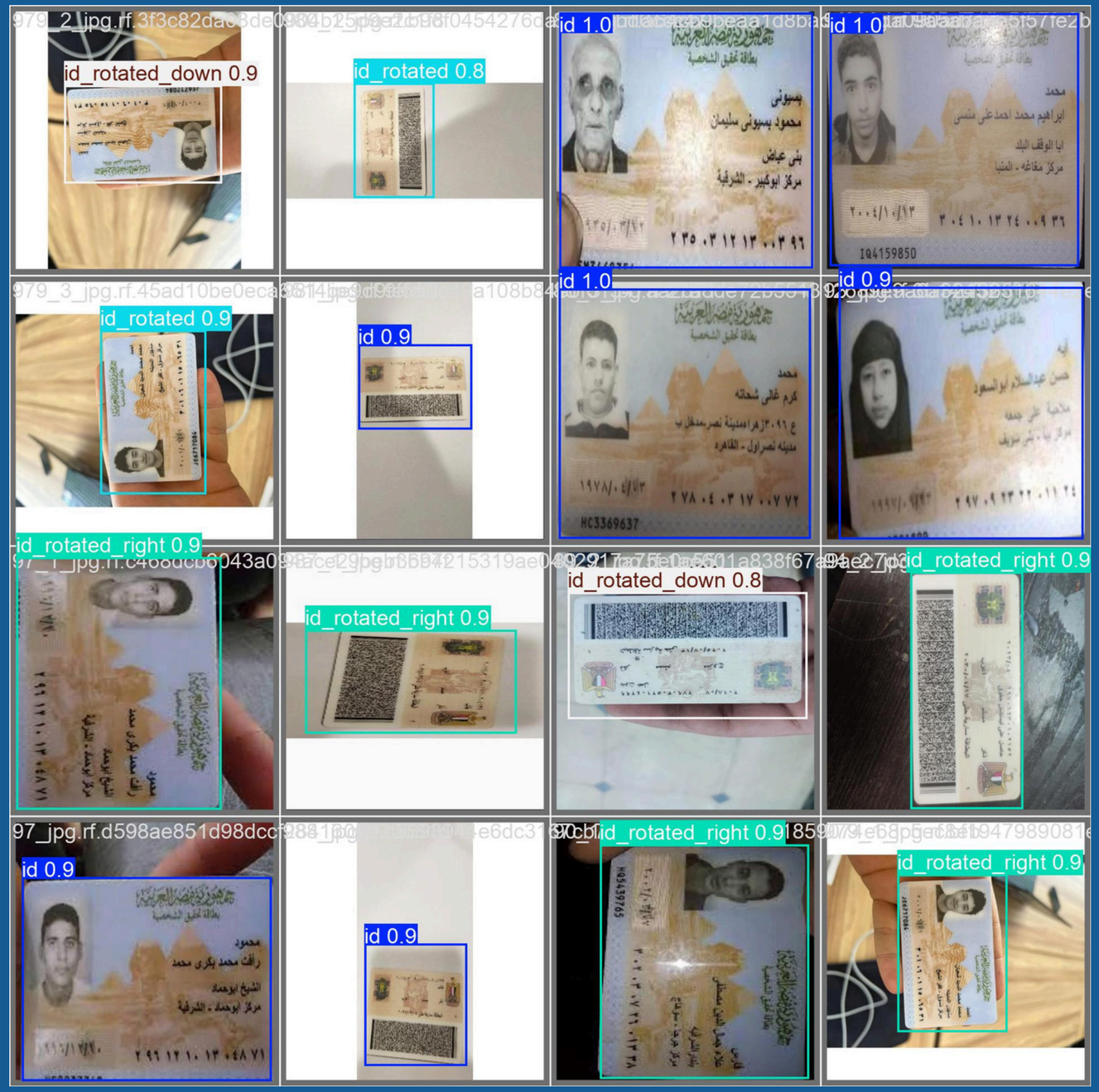


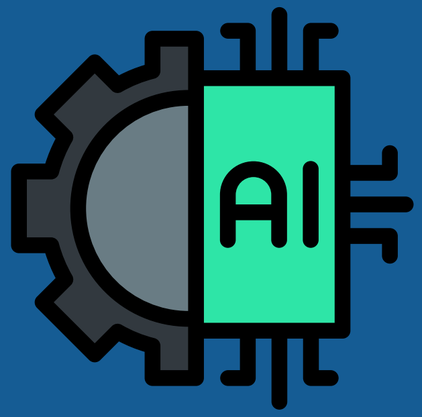
F1-Curve



Predictions

- id_rotated_down: 98%
- id_rotated: 92%
- id: 90%
- id_rotated_right: 92%



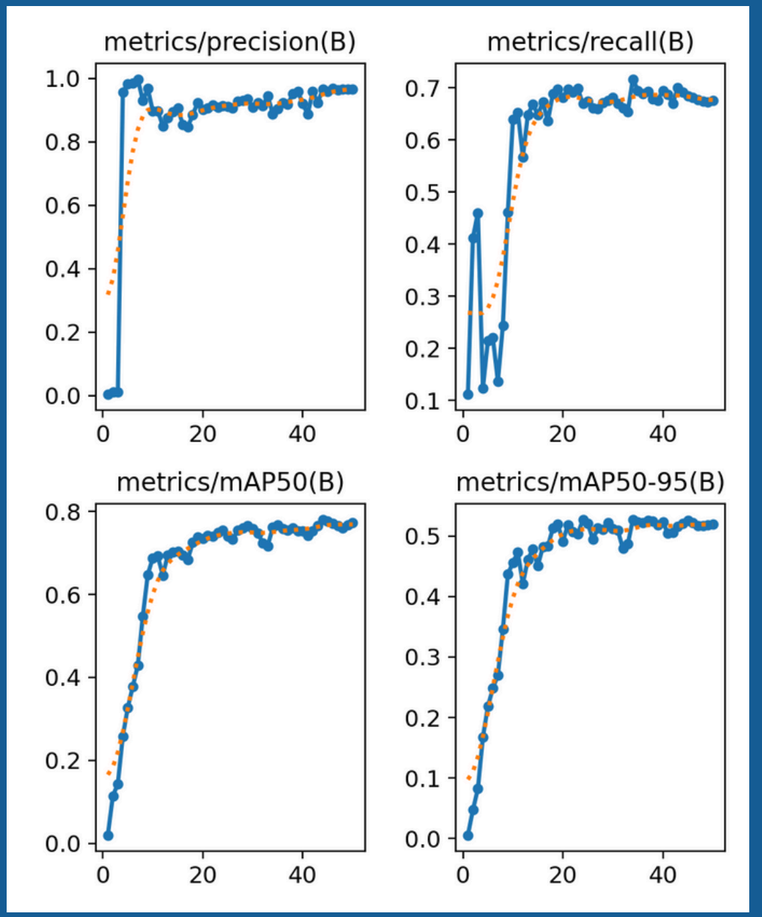


Model 2: Detect Feilds inside ID Card

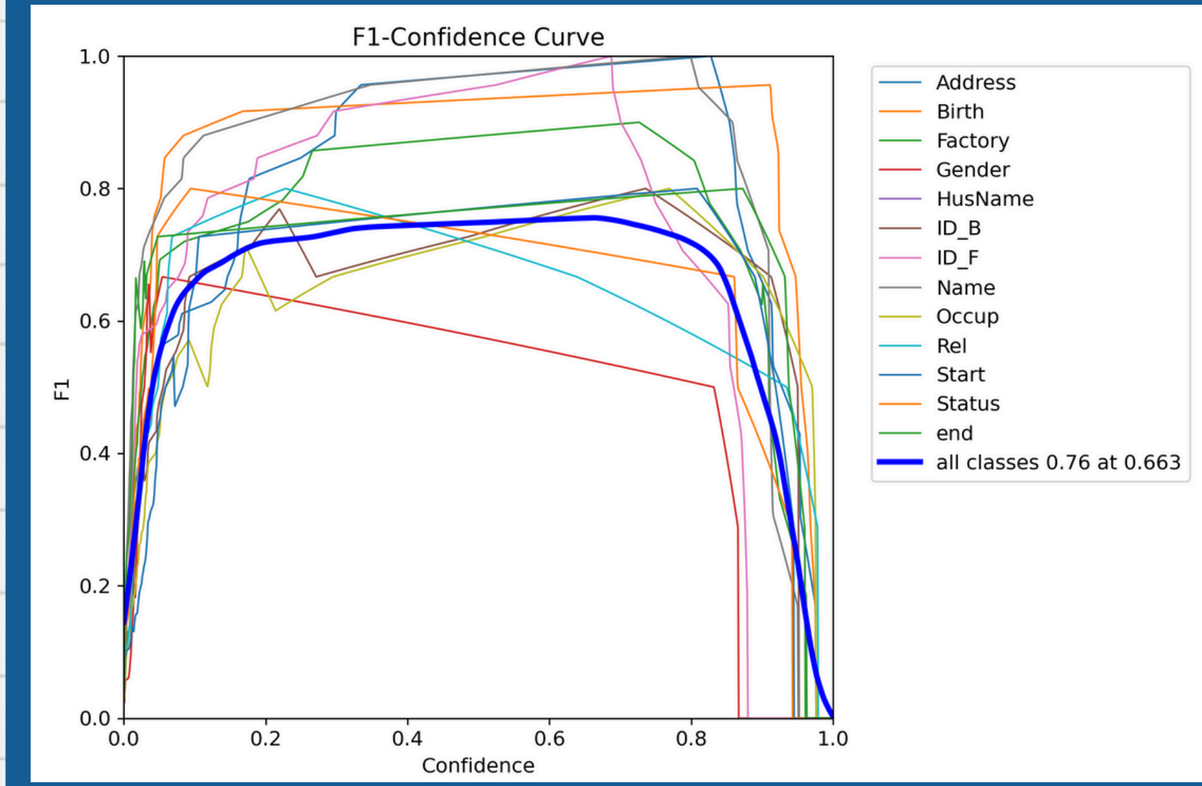
Performance

- Precision (B): 0.96719
- Recall (B): 0.7060
- mAP50 (B): 0.77345

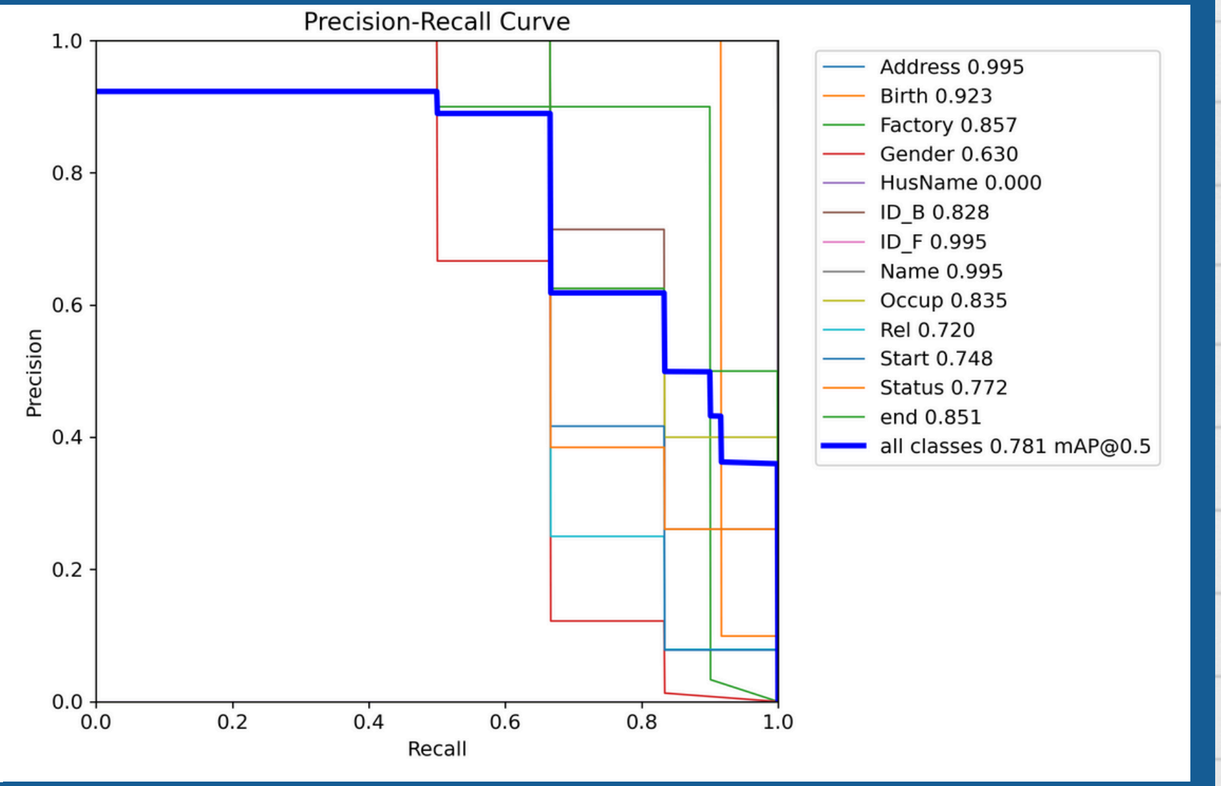
A	B	C	D	E	F	G	H
epoch	time	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)
1	8.81423	2.80502	4.73856	2.25405	0.00368	0.11189	0.01985
2	10.8757	1.96899	4.37742	1.63641	0.01212	0.41224	0.11365
3	12.8917	1.58298	3.88634	1.33118	0.01065	0.46026	0.14372
43	92.5058	1.04318	0.97104	1.06246	0.92344	0.70088	0.76659
44	94.414	1.02939	0.94208	1.05802	0.96624	0.69217	0.78025
45	96.3594	1.01359	0.93279	1.0423	0.96031	0.68517	0.7764
46	98.4796	1.04938	1.04172	1.07853	0.96782	0.68162	0.77214
47	100.398	0.98898	0.91159	1.02056	0.96399	0.67684	0.7666
48	102.316	0.98395	0.92567	1.03883	0.96735	0.67385	0.76101
49	104.313	0.95591	0.89438	1.03052	0.96619	0.67341	0.76738
50	106.178	1.00275	0.90595	1.0468	0.96719	0.67603	0.77345



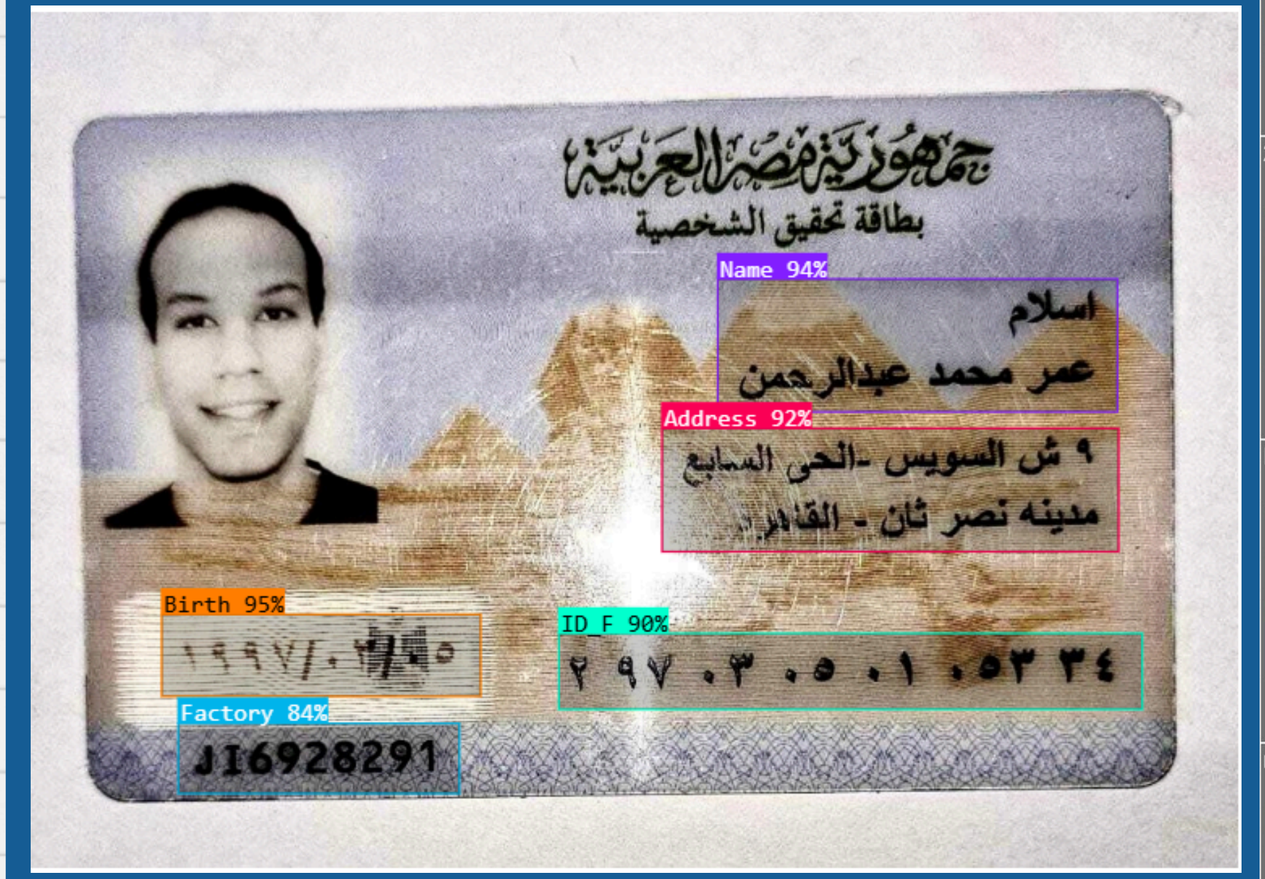
F1-Curve



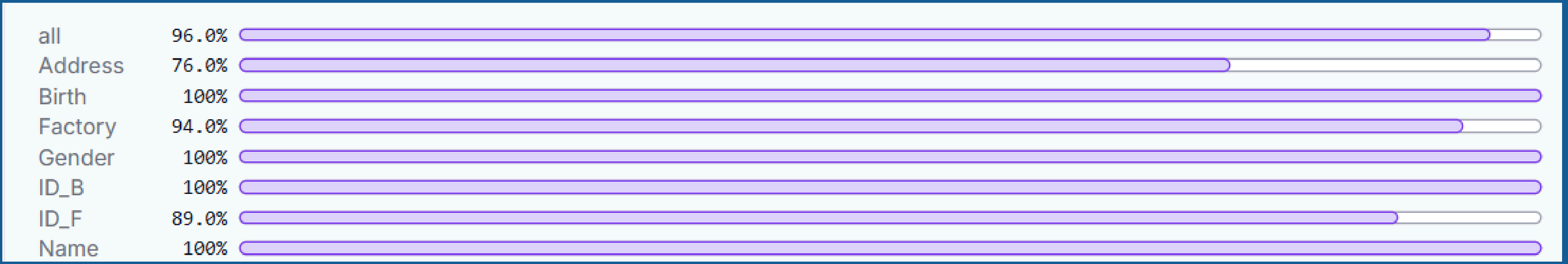
PR Curve



Test Results



🎯 Top Performing Classes



ID Extraction Phases

01

ID detection using YOLO

02

Field detection using YOLO

03

Content extraction using OCR

Phase 1

Phase 2

Phase 1 Summary

- **YOLO Model 1:** ID Detection

This model detects and localizes the National ID card within a larger image, especially useful if the ID is placed on a cluttered background or captured from a camera.

- **YOLO Model 2:** Field Detection

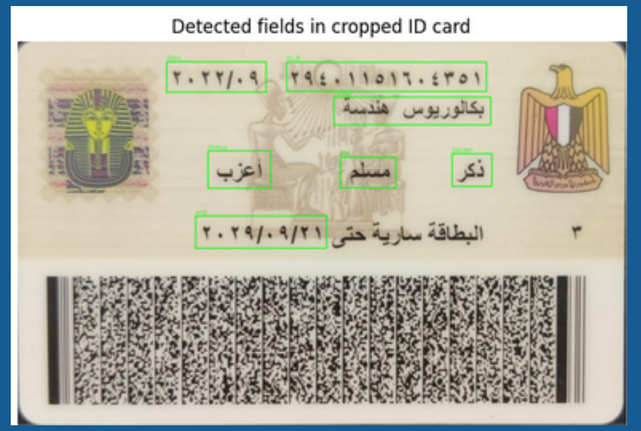
After cropping the ID, we feed it to a second YOLO model trained to detect individual fields on the card: national ID number, name, date of birth ,gender ..etc.

Phase 1 Final Results

This two-step detection helps us isolate only the relevant parts of the ID for further processing

National ID front sample

National ID back sample



READY FOR PHASE 2

🧾 Why OCR is Needed

To **automatically read the text content** from the Egyptian National ID provided during voter registration,
so we can **extract and verify** key information such as:

- 🆔 National ID Number
- 👨‍💼 Full Name
- 📅 Date of Birth

This ensures the user-provided data matches the official ID, enabling a secure and automated identity verification process.

OCR and Field Extraction

After detecting the National ID and its individual fields using our two-stage YOLO pipeline, we crop each detected field and pass it to our OCR module for text recognition.

We used **EasyOCR**, a deep learning–based text recognition model that supports both Arabic and English scripts. This was essential for accurately extracting the ID content, especially the Arabic numbers, and converting them into English digits to match the format that will be provided by the user during the voting process.

Why EasyOCR?

✅ Multilingual Support

- Supports both Arabic and English, which is essential for Egyptian National IDs.

✅ Deep Learning-Based

- Uses CRNN + CTC decoding, providing high accuracy for real-world images with mixed fonts and noisy backgrounds.

✅ No Training Required

- Pretrained and ready to use — saved time and effort compared to building a custom OCR from scratch.

Why EasyOCR?

✅ Field-Level Flexibility

- Performs well on small cropped fields, which matches our YOLO-based pipeline.

✅ Lightweight and Easy to Integrate

- Simple Python API that fits easily into our preprocessing and post-processing pipeline.

OCR Workflow

01

ID detection using YOLO

02

Field detection using YOLO

03

Content extraction using OCR

01

Preprocessing

02

Apply EasyOCR model

03

Post-Processing

04

Final Output

Preprocessing for OCR Accuracy

Before feeding the cropped fields into the OCR model, we applied several preprocessing steps to improve accuracy:

- **Grayscale conversion** to simplify the image and focus on text.
- **Inverse thresholding** to enhance contrast between text and background.
- **Resizing** the image to make small text more readable for OCR.
- **Noise reduction** by simplifying the image structure.

These steps significantly improve OCR performance and ensure the extracted data is clean and usable.

Preprocessing for OCR Accuracy

```
14%|          | 1/7 [00:04<00:24,  4.15s/it]

Image: ID_B_0.jpg
Original Text: ٢٩ ٤ ٠ ١ ١ ٥ ١ ٦ ٠ ٤٣٥١
Extracted Numbers: 29401151604351
```

```
29%|          | 2/7 [00:06<00:14,  2.91s/it]

Image: Start_3.jpg
Original Text: ٢ ٠ ٢ ٢١٠ ٩
Extracted Numbers: 2022/09
```

```
57%|          | 4/7 [00:11<00:08,  2.71s/it]

Image: Occup_5.jpg
Original Text: بكالوريوس هندسة
```

Post-Processing for OCR Accuracy

We applied custom post-processing techniques to improve text extraction quality:

- Converted Arabic digits to English
- Extracted clean numeric values (e.g., NID, dates)
- Removed unwanted symbols and spaces
- Fixed OCR errors like slashes misread as "1"

This approach helps us accurately extract numeric fields for use in our e-voting system.

FINAL RESULTS

OCR Final Results

This table is generated automatically after running the full pipeline on a batch of ID images.
Each field is detected, cropped, preprocessed, and passed through OCR.
The result is then **cleaned and normalized** to ensure it’s accurate and consistent then **saved as CSV file.**

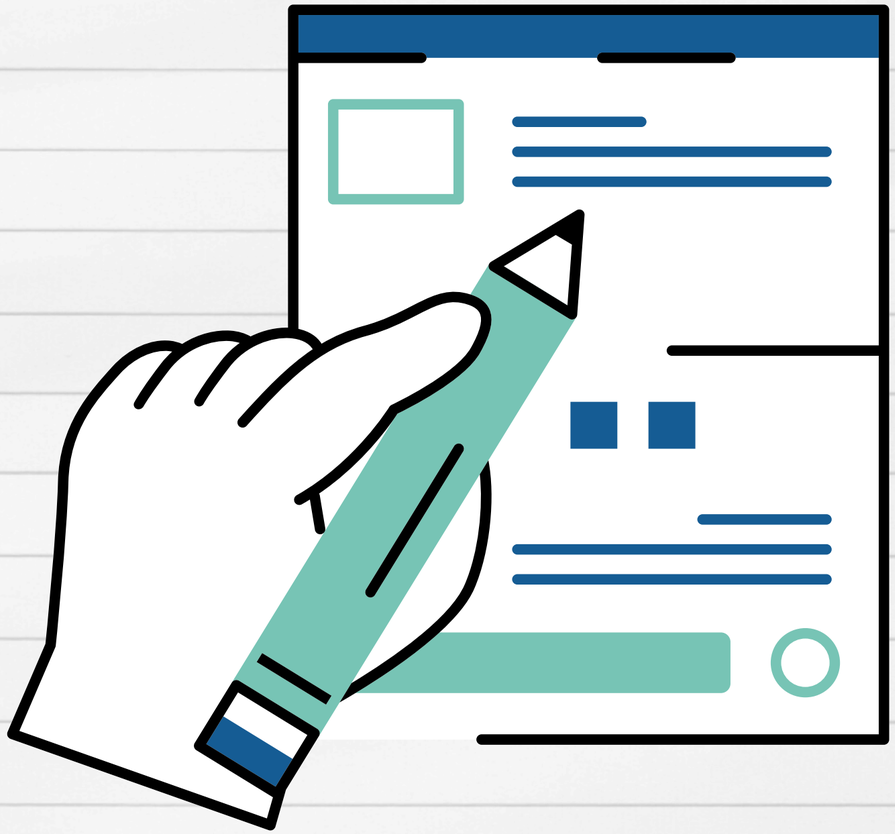
	image_id	original_text	extracted_numbers
0	ID_B_0	٢٩٤٠١١٥١٦٠٤٣٥١	29401151604351
1	Start_3	٢٠٢٢١٠٩	2022/09
2	Status_4	اعزب	
3	Occup_5	بكالوريوس هندسة	
4	Rel_2	مسلم	3
5	Gender_6	ذكر	26
6	end_1	٢٠٢٩١٠٩٢١	2029/09/21

Focused Extraction: Why Only NID?

- Our OCR model was initially built to extract **all key fields** from Egyptian ID cards: Name, Date of Birth, Gender, Job, and the National ID Number (NID).
- After testing the full extraction pipeline, we decided to **focus solely on the NID** for the next stages of development.
- The NID is a **unique and official identifier** used for identity verification and is essential for secure voter linking in the blockchain.
- This focus allowed us to improve OCR accuracy, simplify postprocessing, and ensure a cleaner integration with the blockchain voting system.
- By targeting only the NID, we optimized both performance and security in the overall e-voting workflow.

07

FAKE ID DETECTION



ID CARD VERIFICATION MODEL

- **Purpose:**
  - Detects fake IDs by analyzing images for signs of tampering using AI and forensics techniques (ELA & LBP).
- **Why It Matters:**
  - Used for identity theft & fraud detection .
  - Spots what humans miss under time pressure.
  - Figures out hidden signs of editing on it's own.

PREPROCESSING

we use two forensic techniques:

1. **Error Level Analysis (ELA):** shows parts of an image that might have been changed by showing differences in compression quality.
    - Saves images again at 90% quality.
    - Compares with the original to find spots that don't match.
    - Amplifies differences by 15× so edited parts look brighter.

ELA acts like a tool that catchers Photoshop edits.

PREPROCESSING

2. Local Binary Patterns (LBP): looks at textures in a image to spot signs of editing.

- Analyzes neighborhood pixel relationships.
- Encodes texture patterns as numerical features.
- Detects changes in text style's or background pattern's.

LBP works as a fingerprint reader but for image textures.

PREPROCESSING

Feature	ELA	LBP
Detects	Digital tampering	Physical tampering
Best for	Edited parts in digital images	Printed/Scan quality issues
strengths	Catches tiny digital edits	Good for paper or scanned id's
Outputs	Heatmap of edits	Texture pattern map

MODEL ARCHITECTURE

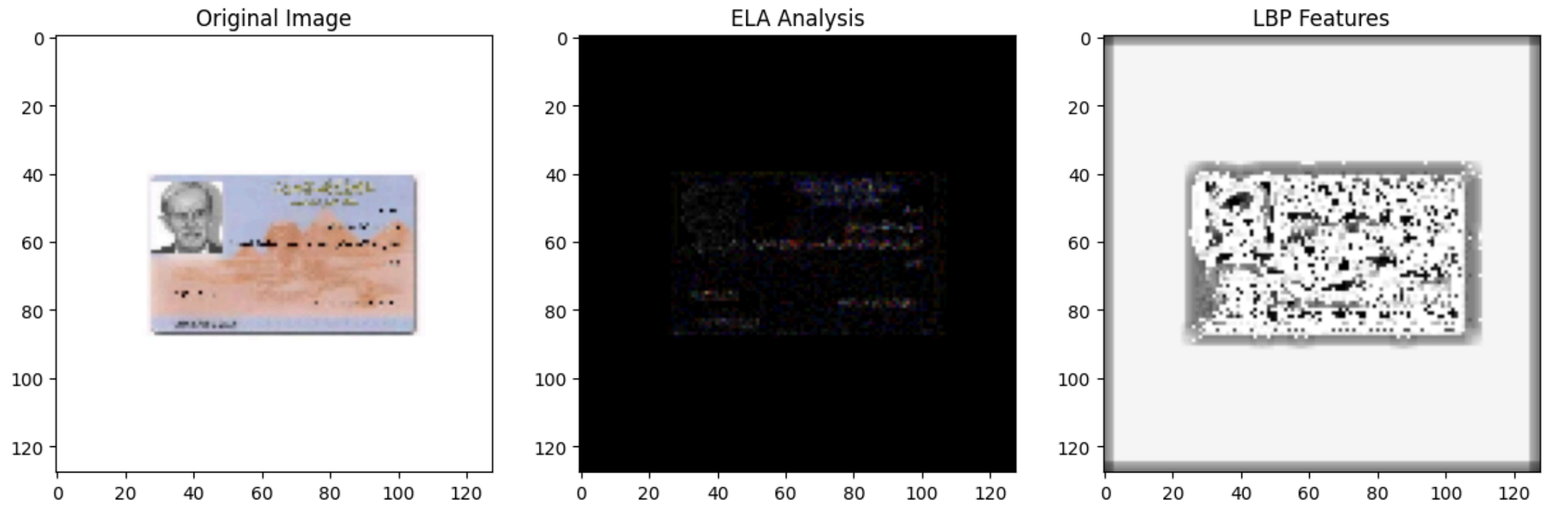
Three-Branch Hybrid Network:

- **Original Image Branch:** EfficientNetB0,cause it recognizes fine details.
- **ELA Branch:** Processing layer, it finds editing traces.
- **LBP Branch:** Processing layer,it catches physical tampering.
- **Feature Fusion from all three branches.**

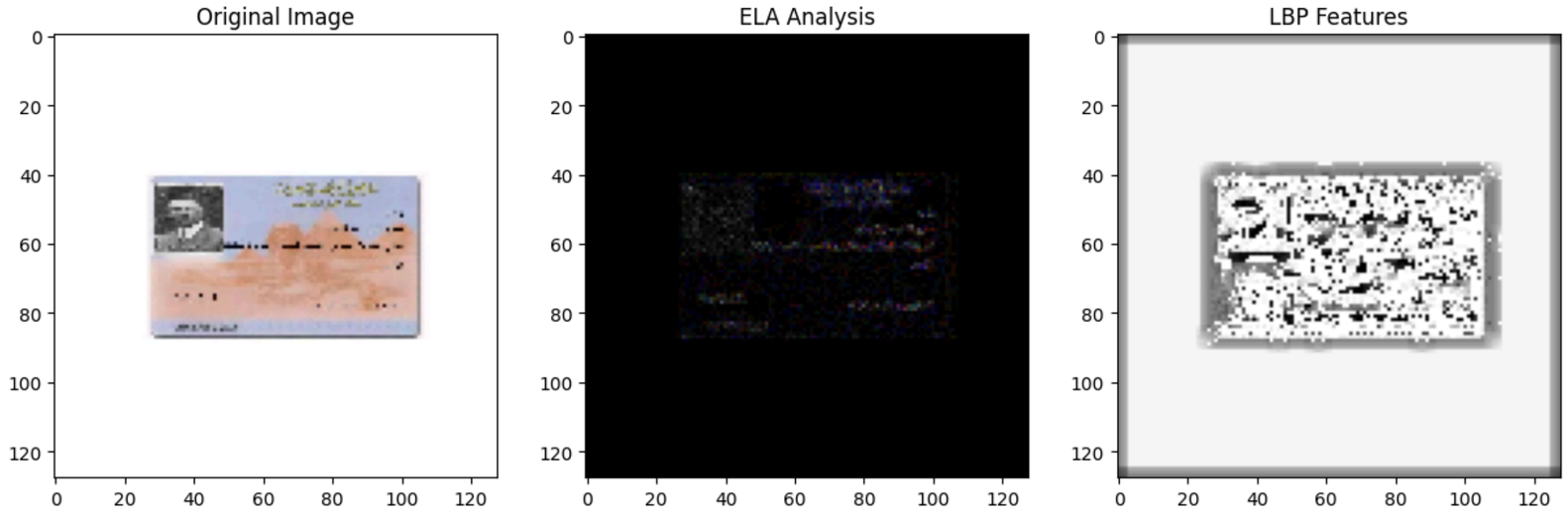
```
- val_recall: 0.9953 - learning_rate: 3.0000e-06
Epoch 2/8
437/437 ━━━━━━━━━━━━━━━━━━━━ 221s 496ms/step - accuracy: 0.8683 - loss: 0.3294 - precision: 0.8075 - recall: 0.9403 - val_accuracy: 0.9978 - val_loss: 0.0521 - val_precision: 1.0000
- val_recall: 0.9953 - learning_rate: 3.0000e-06
Epoch 3/8
437/437 ━━━━━━━━━━━━━━━━━━━━ 206s 471ms/step - accuracy: 0.9654 - loss: 0.1596 - precision: 0.9538 - recall: 0.9725 - val_accuracy: 0.9981 - val_loss: 0.0274 - val_precision: 1.0000
- val_recall: 0.9958 - learning_rate: 3.0000e-06
Epoch 4/8
437/437 ━━━━━━━━━━━━━━━━━━━━ 201s 461ms/step - accuracy: 0.9920 - loss: 0.0901 - precision: 0.9929 - recall: 0.9897 - val_accuracy: 0.9983 - val_loss: 0.0170 - val_precision: 1.0000
- val_recall: 0.9963 - learning_rate: 3.0000e-06
Epoch 5/8
437/437 ━━━━━━━━━━━━━━━━━━━━ 199s 457ms/step - accuracy: 0.9964 - loss: 0.0559 - precision: 0.9974 - recall: 0.9947 - val_accuracy: 0.9985 - val_loss: 0.0118 - val_precision: 1.0000
- val_recall: 0.9967 - learning_rate: 3.0000e-06
Epoch 6/8
437/437 ━━━━━━━━━━━━━━━━━━━━ 187s 430ms/step - accuracy: 0.9979 - loss: 0.0390 - precision: 0.9989 - recall: 0.9965 - val_accuracy: 0.9985 - val_loss: 0.0087 - val_precision: 1.0000
- val_recall: 0.9967 - learning_rate: 3.0000e-06
Epoch 7/8
437/437 ━━━━━━━━━━━━━━━━━━━━ 169s 388ms/step - accuracy: 0.9989 - loss: 0.0270 - precision: 0.9997 - recall: 0.9980 - val_accuracy: 0.9985 - val_loss: 0.0068 - val_precision: 1.0000
- val_recall: 0.9967 - learning_rate: 3.0000e-06
Epoch 8/8
437/437 ━━━━━━━━━━━━━━━━━━━━ 149s 342ms/step - accuracy: 0.9989 - loss: 0.0205 - precision: 1.0000 - recall: 0.9976 - val_accuracy: 0.9985 - val_loss: 0.0056 - val_precision: 1.0000
- val_recall: 0.9967 - learning_rate: 3.0000e-06
```

- The model gets accuracy of 99.9% on train and 99.9%on validation

```
Testing image: 0.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
Prediction: FAKE (100.0% confidence)
```

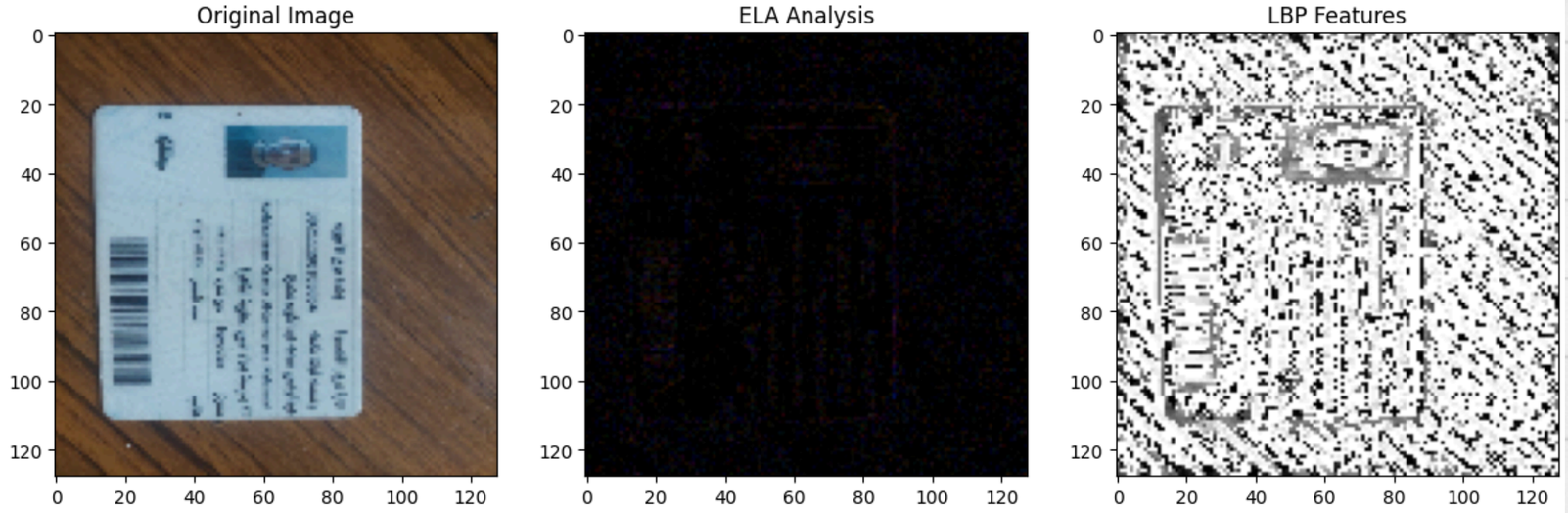



```
Testing image: 1.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
Prediction: FAKE (100.0% confidence)
```

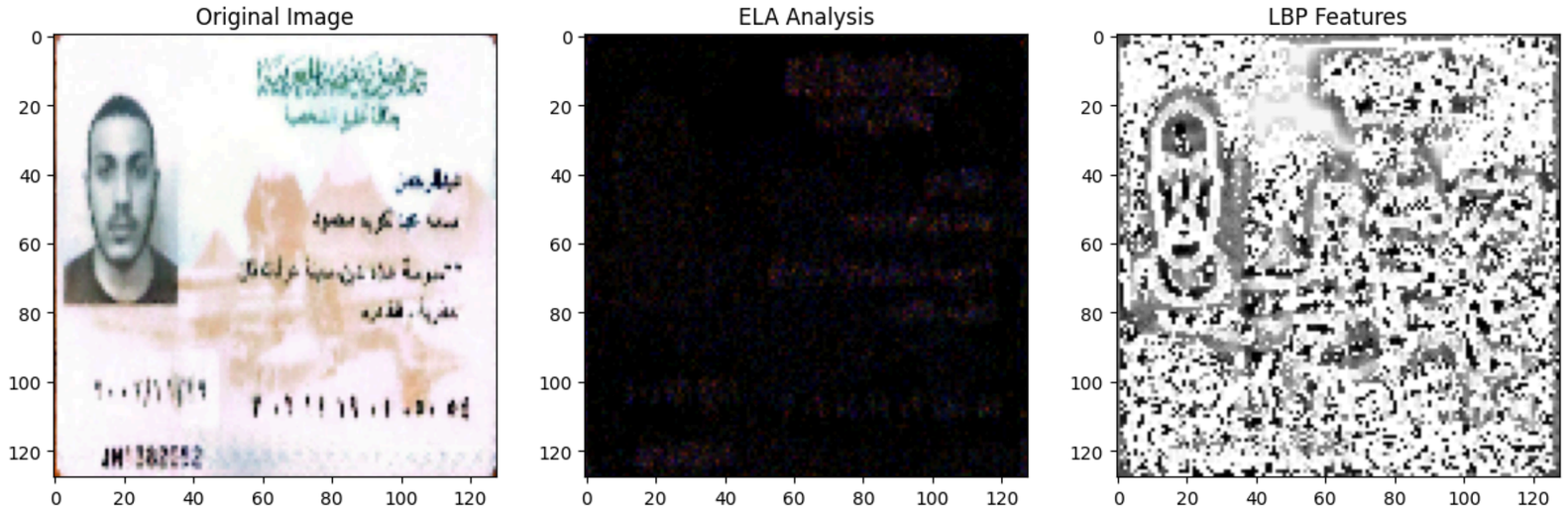


- And here are samples of the test on fake id cards

```
Testing image: WhatsApp Image 2025-04-12 at 22.59.56_cfb2df2b.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
Prediction: REAL (100.0% confidence)
```



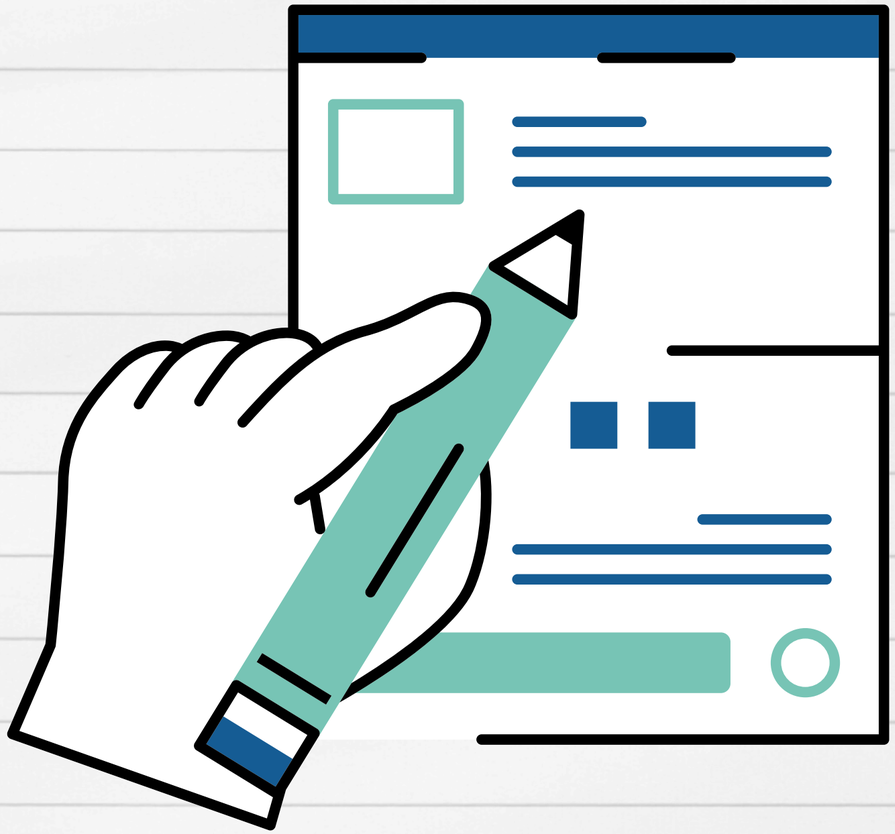

```
Testing image: WhatsApp Image 2025-03-25 at 00.47.22_daab1142.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
Prediction: REAL (78.7% confidence)
```



- And here are samples of real id cards

09

BLOCKCHAIN



ZK-SYNC

- **Layer 1 Security:** zkSync inherits the security guarantees of Ethereum Layer 1, meaning the system does not sacrifice any security while benefiting from scalability.

- **Security and Immutability:** zkSync uses zero-knowledge proofs to ensure that all transactions are secure and immutable, meeting the system’s requirement for tamper-proof voting records.

- **Smart Contract Support:** zkSync is compatible with Solidity, enabling seamless development and integration of voting-related smart contracts.

ZK-SYNC

- **Account Abstraction:** zkSync simplifies the user experience by handling transaction fees, signatures, and wallet logic behind the scenes, making blockchain voting accessible to everyone.
    - Transaction Sponsorship
- **Scalability:** Capable of handling thousands of transactions per second.
- **Cost-Efficiency:** zkSync significantly reduces transaction fees up to 95-98% compared to Ethereum Layer 1, making it suitable for high-volume voting scenarios.

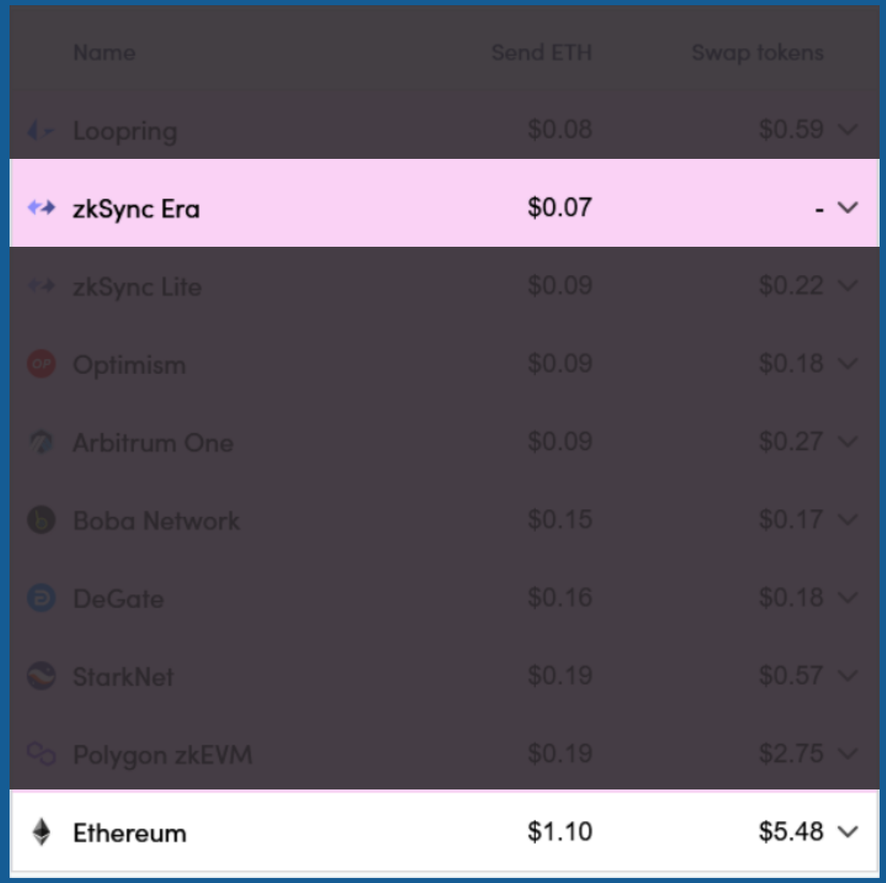
TRANSACTION FEES

zk-Sync
VS
Ethereum Mainnet

Name	Send ETH	Swap tokens
Loopring	$0.08	$0.59
zkSync Era	$0.07	-
zkSync Lite	$0.09	$0.22
Optimism	$0.09	$0.18
Arbitrum One	$0.09	$0.27
Boba Network	$0.15	$0.17
DeGate	$0.16	$0.18
StarkNet	$0.19	$0.57
Polygon zkEVM	$0.19	$2.75
Ethereum	$1.10	$5.48

TRANSACTION FEES

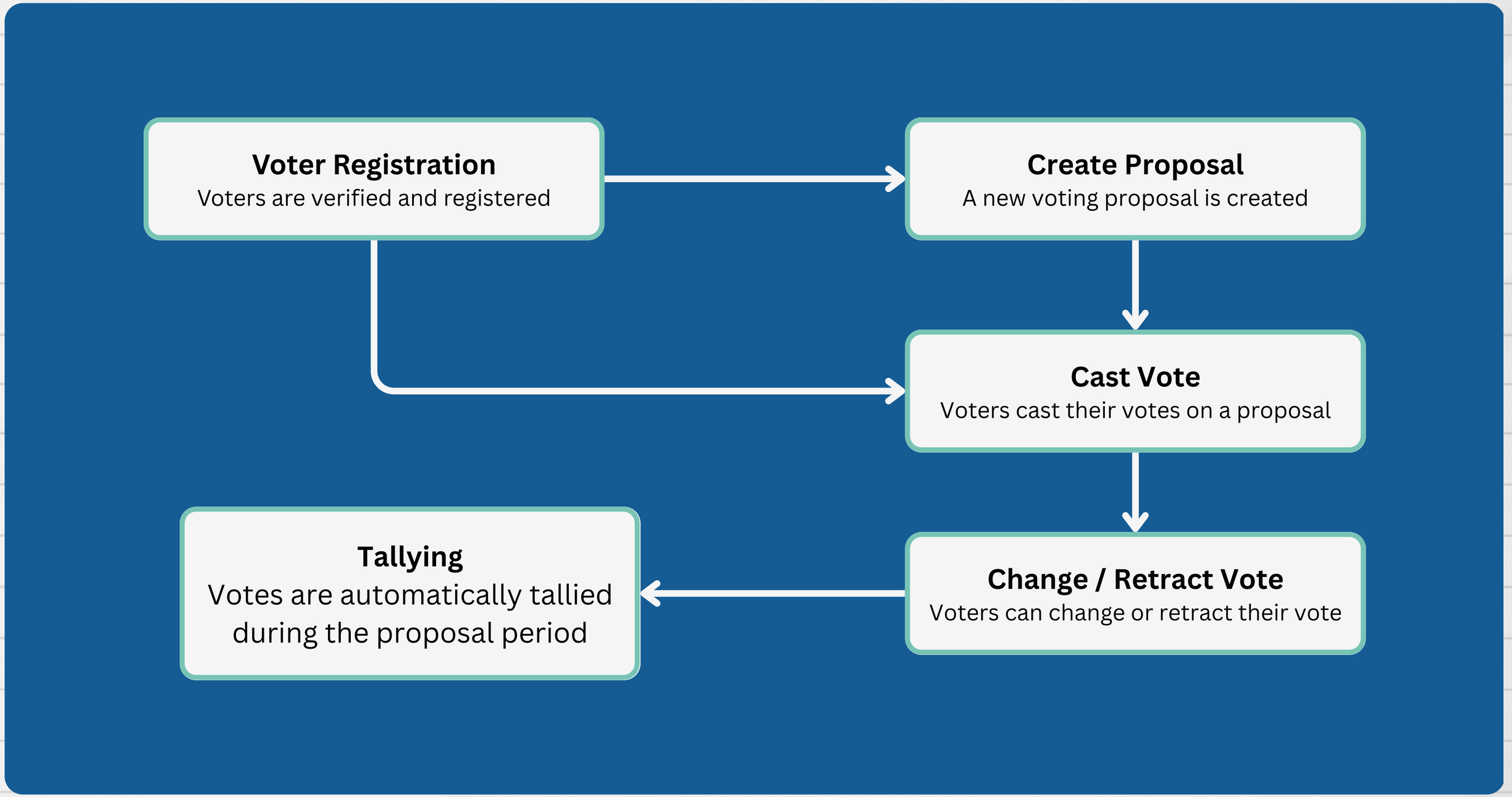
zk-Sync
VS
Ethereum Mainnet



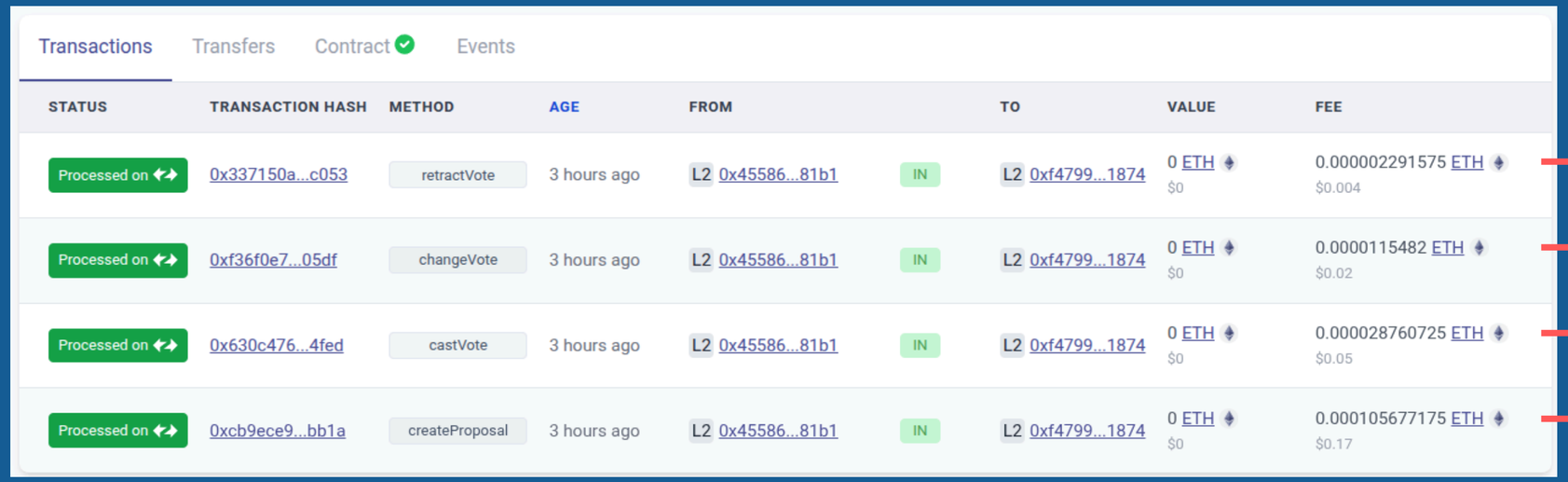
Name	Send ETH	Swap tokens
Loopring	$0.08	$0.59
zkSync Era	$0.07	-
zkSync Lite	$0.09	$0.22
Optimism	$0.09	$0.18
Arbitrum One	$0.09	$0.27
Boba Network	$0.15	$0.17
DeGate	$0.16	$0.18
StarkNet	$0.19	$0.57
Polygon zkEVM	$0.19	$2.75
Ethereum	$1.10	$5.48

Implementation

- User Registration & Verification
- Creating Proposals
- Deleting Proposals
- Casting Votes
- Retracting Votes
- Changing Votes


Voter Registration
Voters are verified and registered
Create Proposal
A new voting proposal is created
Cast Vote
Voters cast their votes on a proposal
Change / Retract Vote
Voters can change or retract their vote
Tallying
Votes are automatically tallied during the proposal period

Deployment: zKSync



Transactions | Transfers | Contract | Events

STATUS	TRANSACTION HASH	METHOD	AGE	FROM		TO	VALUE	FEE
Processed on	0x337150a...c053	retractVote	3 hours ago	L2 0x45586...81b1	IN	L2 0xf4799...1874	0 ETH $0	0.000002291575 ETH $0.004
Processed on	0xf36f0e7...05df	changeVote	3 hours ago	L2 0x45586...81b1	IN	L2 0xf4799...1874	0 ETH $0	0.0000115482 ETH $0.02
Processed on	0x630c476...4fed	castVote	3 hours ago	L2 0x45586...81b1	IN	L2 0xf4799...1874	0 ETH $0	0.000028760725 ETH $0.05
Processed on	0xcb9ece9...bb1a	createProposal	3 hours ago	L2 0x45586...81b1	IN	L2 0xf4799...1874	0 ETH $0	0.000105677175 ETH $0.17

→ $ 0.003

→ $ 0.019

→ $ 0.048

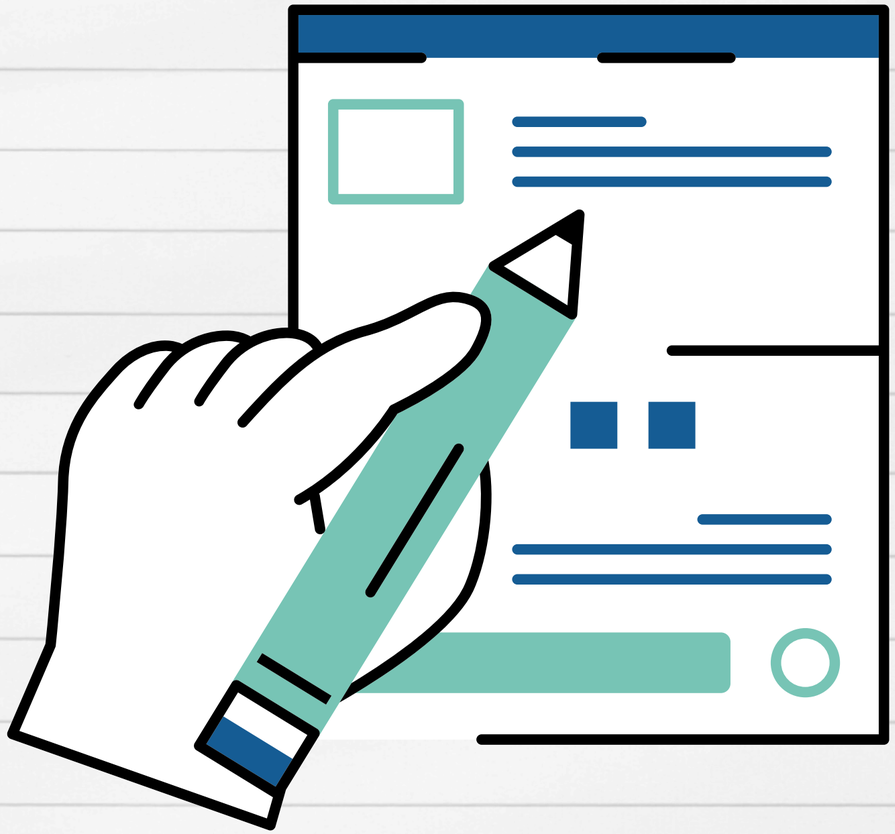
→ $ 0.177

Avg cost per Transaction

$ 0.06

10

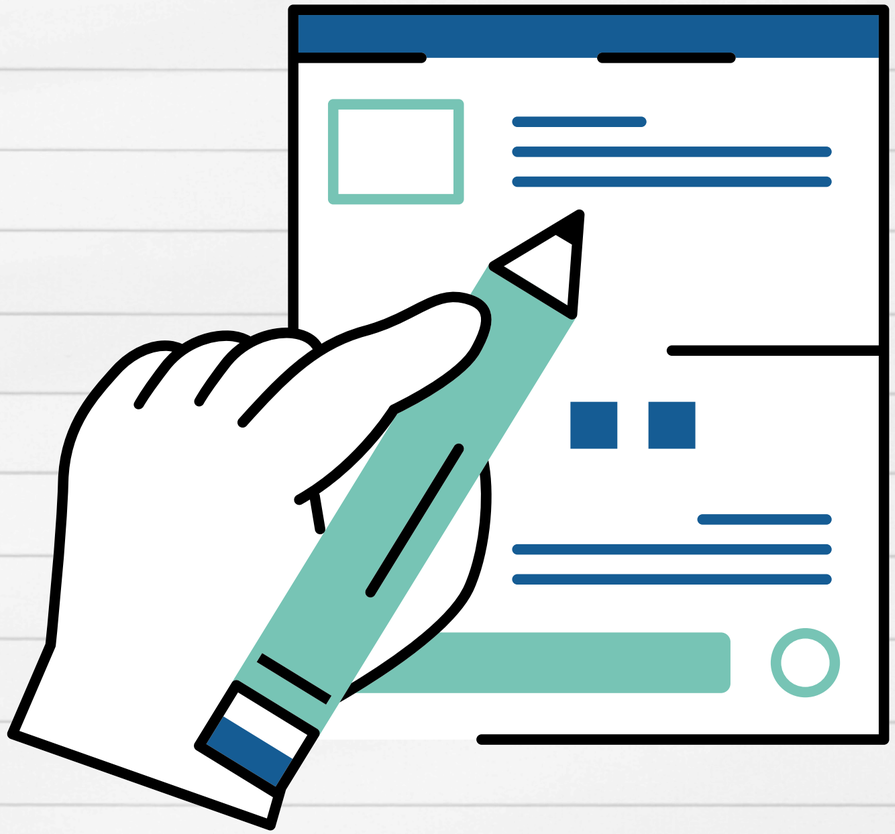
FRONTEND



DEMO TIME

11

CONCLUSION



PROJECT SUMMARY & IMPACT

- Developed a secure, AI-powered e-voting system using blockchain.
- Integrated multiple technologies:
  - Face Verification
  - Fake ID Detection
  - OCR for NID Extraction
  - Object Detection
  - Dynamic Web Interface
  - zkSync Blockchain
- Ensures:
  - Identity Security
  - Vote Integrity
  - Scalability & Accessibility

WHAT WE ACCOMPLISHED

- Built a full pipeline from user verification to vote recording.
- Successfully extracted and verified National ID Numbers using OCR.
- Deployed facial recognition and fake ID detection for double-layered security.
- Integrated zkSync blockchain to store votes with immutability and transparency.
- Delivered a user-friendly web interface that hides backend complexity.

FUTURE ENHANCEMENTS

- **OCR Error Handling**

Add a fallback flow to prompt the user to re-upload ID if OCR fails due to blur, lighting, or image quality.

- **Liveness Detection**

Implement blink or head-movement detection to prevent face spoofing using photos or videos.

- **Offline Voting with Sync**

Support secure offline voting with sync once the user reconnects to the internet.

FUTURE EXPANSION

- **Multi-modal Biometric Verification**

Introduce fingerprint or voice recognition for more flexible and secure user verification.

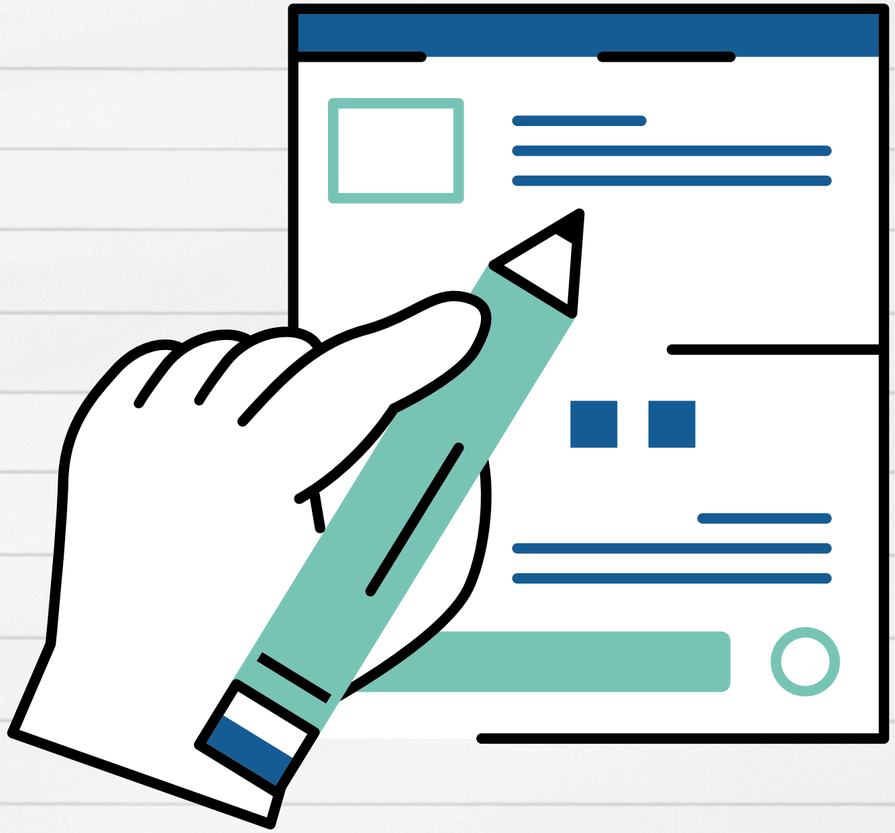
- **Integration with National Databases**

Real-time ID verification via government databases for enhanced accuracy and trust.

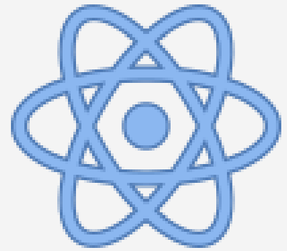
- **Broader Use Cases**

Extend the system to digital identity services like e-passport control, subsidy distribution, and more.

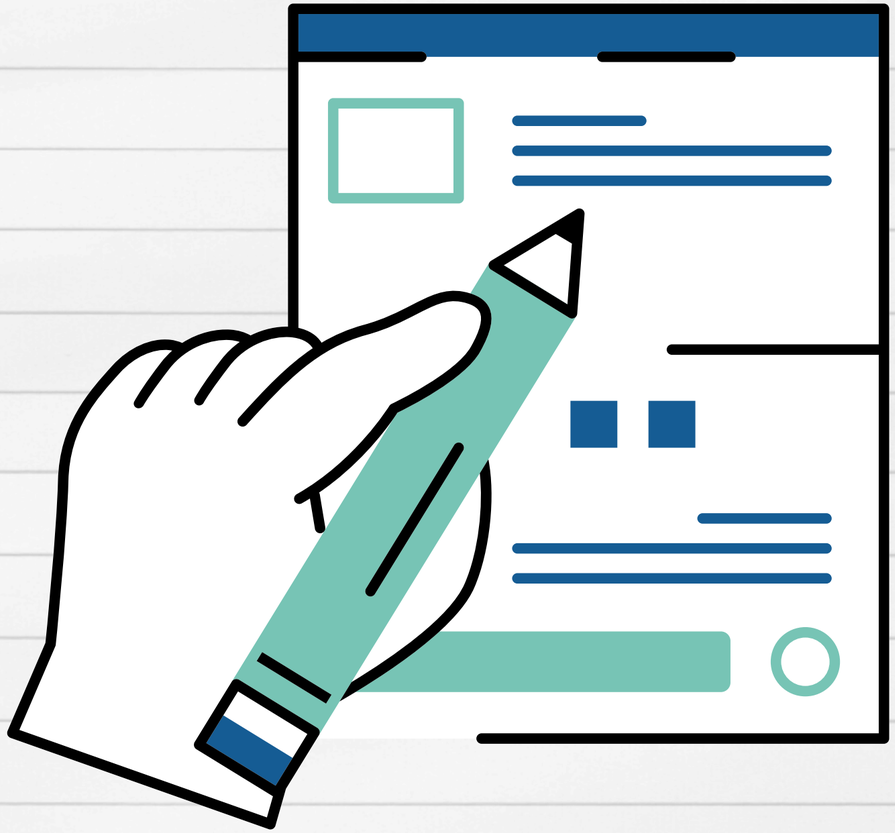
TOOLS



TOOLS

ZK-Sync	Solidity	Python	React
			

REFERENCES



REFERENCES

[1] “VGGFace2: A dataset for recognising faces across pose and age” Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi and Andrew Zisserman Visual Geometry Group, Department of Engineering Science, University of Oxford {qiong,lishen,weidi,omkar,az}@robots.ox.ac.uk

[2] “Siamese Neural Networks for One-shot Image Recognition” Gregory Koch, Richard Zemel , Ruslan Salakhutdinov, Department of Computer Science, University of Toronto. Toronto, Ontario, Canada. [https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf]

[3] “FaceNet: A Unified Embedding for Face Recognition and Clustering” Florian Schroff, Dmitry Kalenichenko , James Philbin, Google Inc. [https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Schroff_FaceNet_A_Unified_2015_CVPR_paper.pdf]

[4] “Learning Face Representation from Scratch” Dong Yi, Zhen Lei, Shengcai Liao and Stan Z. Li Center for Biometrics and Security Research & National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences (CASIA)

[5] “Learning Robust Deep Face Representation” Xiang Wu University of Science and Technology Beijing Beijing, China

REFERENCES

[6]“Voting System Using Blockchain (Face Recognition)” Gaddam Harsha Vardhan, Swapnil Shah, Vanshika Gupta, Rohithreddy.B.C, Tanya Bisht

[6] “E-Voting using Blockchain: Moving Away from the Ballot Paper” Irvine Mutandiwa, Calvin P Mugauri, Maronge Musara

[7] “Blockchain-Based Electronic Voting: A Secure and Transparent Solution” Bruno Miguel Batista Pereira, José Manuel Torres, Pedro Miguel Sobral, Rui Silva Moreira, Christophe Pinto de Almeida Soares, Ivo Pereira

[8] “Blockchain-Based E-Voting System with Face Recognition” V. Sathya Preiya, V. D. Ambeth Kumar, R. Vijay, Vijay K., N. Kirubakaran

[9] “A Blockchain-based Electronic Voting System: EtherVote” Achilleas Spanos Department of Informatics and Computer Engineering, University of West Attica Athens, Greece,
Ioanna Kantzavelou Department of Informatics and Computer Engineering, University of West Attica Athens, Greece

[10] “A Privacy-Preserving Blockchain-based E-voting System” Arnab Mukherjee, Souvik Majumdar, Anup Kumar Kolya, Saborni Nandi

THANK YOU VERY MUCH!