International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Online Voting System Based on Face Recognition Using Python Machine Learning.

Mahammednawaj Mulla

Rani Channamma University, Belagavi

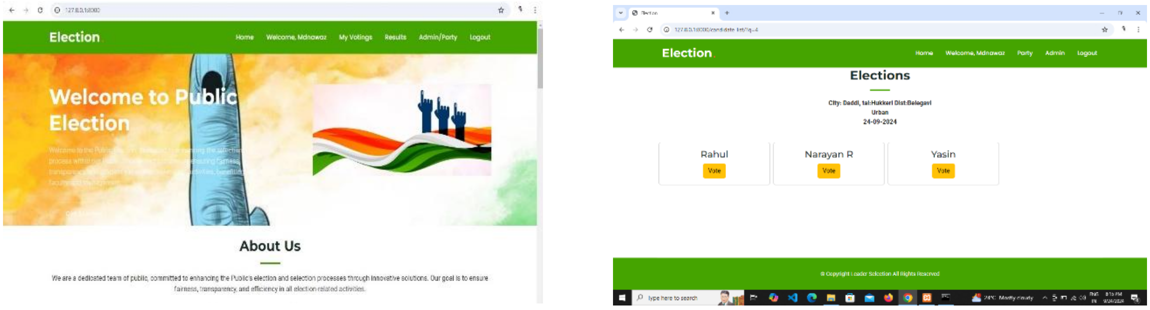
ABSTRAC T

The use of facial recognition and One-Time Password in an online voting system ensures that the correct individual is casting their vote. Initially, the system employs facial recognition to verify if the voter's face matches the one on file, reducing the possibility of fraudulent activities. Subsequently, a unique one-time password is sent to the voter's mobile device, and they must input it to confirm their identity. This dual verification process, combining facial recognition and OTP, enhances the security and dependability of the voting system. The utilization of Python's machine learning tools facilitates the efficient development of this system. Python and machine learning technologies are utilized to create a secure and efficient online voting system based on facial recognition and One-Time Password authentication. This system leverages facial recognition algorithms to authenticate the voter's identity, thereby minimizing the risk of impersonation and fraud. It is further bolstered by OTP verification, adding an additional layer of security to ensure that only authorized voters are participating. Python's machine learning libraries, including OpenCV and TensorFlow, are utilized to implement facial recognition, while OTP functionality strengthens the security of the system. This comprehensive approach promotes a robust, secure, and user-friendly voting process suitable for large-scale elections.

1. Introduction

Traditional voting systems often have problems. Long lines, limited resources, & the risk of fraud can make voting tough. But online voting is here to help! It gives voters a simple & easy way to cast their votes from anywhere. With the use of machine learning (ML) techniques, online voting gets even better. It helps make sure that voter identities are accurate, finds fraud quickly, & keeps elections fair. This online system can boost participation too! It's especially helpful for those who may struggle to reach polling places—like people with mobility issues, disabilities, or those who live far away.With more people using digital systems these days, there's a big chance to improve how we vote. An online voting system that uses face recognition can make things easier and safer. It helps tackle problems like voter fraud, accessibility, & convenience. Traditional ways of voting whether in person or through electronic means have some issues Things like proving who you are, preventing tampering, and handling a lot of voters are tough, especially in big elections. But by adding biometric authentication, especially face recognition, we can solve many of these problems. It offers a dependable way to confirm each voter's identity.

Face recognition technology uses clever machine learning tools. These tools look at and compare facial features to enhance security. This way, only the right people can cast their votes, which stops impersonation and duplicate votes. When you mix this with **blockchain technology**, the voting process becomes even more secure & clear. It helps make sure that everything is locked in place and can be checked.Plus, this system would be great for giving more people the chance to vote from anywhere! This is super helpful for folks in areas where voting places are hard to find or for those with disabilities. Still, there are some things we need to watch out for, like privacy concerns, keeping data safe, and bias in face recognition systems. These are important steps to keep everything fair & trustworthy. So, all in all, using face recognition for online voting could be a really cool way to create easier, safer elections for everyone.



We are a dedicated team of public, committed to enhancing the Public's election and selection processes through innovative solutions. Our goal is to ensure fairness, transparency, and efficiency in all election-related activities.

City: Daddi, tal:Hukkeri Dist:Belegavi
Urban
24-09-2024

© Copyright Leader Selection All Rights Reserved

Figure 1: Online Voting System.

2. RelatedWork

2.1 Previous Research and Implementations

There have been various studies exploring the integration of face recognition technology into online voting systems:

- **2018 Study:** This research combined facial recognition with online voting, discovering methods to enhance voter authentication accuracy [1].
- **2020 Research:** A study focused on the effectiveness of face and fingerprint recognition in preventing election fraud. Trials conducted in parts of Africa and Asia showed promising results[2]
- **2022 Project:** A project developed a prototype of an e-voting system that used facial recognition for security. The system showed great potential, notably reducing risks like impersonation and double voting [3].

2.2 Literature Review:

Online voting systems are digital tools that enable voters to cast their ballots electronically, offering convenience by allowing participation from any location without the need to visit polling stations. However, these systems face significant challenges, particularly in verifying voter identities and preventing fraud. Traditional authentication methods, such as usernames and passwords, are susceptible to attacks and falsification, leading to increased exploration of biometric authentication, particularly facial recognition, to enhance security and reliability in online voting [4, 5]. Biometric systems, which use unique physiological characteristics like fingerprints or facial features, are becoming more prominent in online voting due to their robustness in ensuring secure voter identification [6]. Facial recognition, in particular, has gained popularity for its user-friendly nature, driven by advancements in artificial intelligence (AI). During registration, voters provide personal information, such as their name and email, along with multiple facial images, typically around 20, which are used to create a secure digital "face print" [7, 8].

When voters log in to vote, the system compares a live picture taken with a webcam or smartphone camera to this stored face print using sophisticated algorithms. If the images match, access is granted; otherwise, it is denied [9]. The technology behind facial recognition relies on machine learning and deep learning techniques, particularly Convolutional Neural Networks (CNNs), which analyze distinct facial features through processes like face detection, feature extraction, and matching with stored data [10]. Training involves providing multiple images under various conditions to ensure accurate recognition [11]. To ensure the security of online voting systems incorporating facial recognition, several key aspects must be considered, including data encryption to protect personal information [12], liveness detection to prevent spoofing attacks [13].

3. MaterialsandMethods

Creating an online voting system with face recognition combines lots of different technologies & programming languages. Each part plays its role. This includes machine learning, web development, & security. Let's take a look at the main tools and languages used here:

3.1 Programming Languages:

Python: Python is recognized for its simplicity and readability, is a versatile and high-level programming language suitable for both beginners and experts. It accommodates various programming paradigms like procedural, object-oriented, and functional programming. Python finds extensive usage in machine learning, artificial intelligence, and scientific computing, and it is supported by comprehensive libraries and frameworks such as Django for web development and Pandas for data analysis. The language benefits from an active community that consistently contributes to a wealth of resources, tutorials, and tools, promoting innovation and collaboration among developers globally.Popular Libraries Include:

- **face_recognition**: A simple library built on dlib that helps recognize faces.
- **OpenCV**: Great for things like detecting faces and capturing videos.
- **Flask or Django**: They help create the web interface & REST API.

JavaScript: It works on the client side to create a lively user interface. Frameworks like React or Vue.js make the front-end experience even better. It can also work with HTML5 to grab images from the webcam right in the browser.

HTML/CSS: These are the bases for building the front interface of our website.CSS frameworks like Bootstrap help design responsive layouts that look good on any screen size.

3.2 Face Recognition:

Face Recognition: This makes use of deep learning models to spot & recognize faces in photos or videos. The face_recognition library (which is based on dlib and ResNet models) provides ready-made models for detecting and encoding faces.

Image Processing: OpenCV, also known as Open Source Computer Vision Library, is used for capturing images, manipulating them (such as changing their size), and identifying faces in live settings. For more advanced machine learning assignments, tools like scikit-learn come in handy, particularly for developing specific classifiers. If you want to create personalized models for facial recognition, you could also think about employing TensorFlow or PyTorch.

3.3 Web Development and API: When developing the online voting system, one can choose between using Flask or Django for the backend development. Flask is suitable for small to medium applications and is known for its user-friendly nature, while Django provides a more intricate framework with integrated tools for managing databases and authenticating users. Both frameworks enable the creation of a seamless web experience, efficiently handling user requests and integrating with the facial recognition system. For the front end, one can utilize JavaScript and frameworks like React, Vue.js, or Angular to craft attractive and user-friendly interfaces. JavaScript is especially beneficial due to its capacity to access webcams and easily manage image uploads from users, thereby improving the overall functionality of the platform.

3.4 Database and Storage:

- **SQL Databases:** SQLite, MySQL, or PostgreSQL work well for holding user data, face encodings, & voting records.
- **NoSQL Databases:** MongoDB is great when you need flexible data structures like user profiles or logs.
- **File Storage:** For storing images and models safely, you might think about using cloud storage like AWS S3, Google Cloud Storage, or Azure Blob Storage.

3.5 Security and Encryption:

Data security can be ensured by incorporating data encryption with libraries like Cryptography in Python to safeguard crucial information, such as user images and voting records. Additionally, it is important to always use HTTPS (TLS/SSL) to maintain secure communication between clients and servers. To enhance authentication and authorization processes, it is recommended to use JWT (JSON Web Tokens) or OAuth for safe user authentication. Moreover, it is crucial to integrate anti-spoofing measures by utilizing techniques that verify liveness using OpenCV or implementing intelligent neural network models capable of identifying counterfeit images.

3.6 CNN ALGORITHM:

CNN Algorithm is a particular kind of deep learning algorithm that is particularly useful for processing grid-like input, like photographs, is the convolutional neural network (CNN). Thanks to their capacity to automatically recognize and learn essential properties from input data, CNNs have become extremely useful for tasks like object detection and picture recognition, among many other applications.

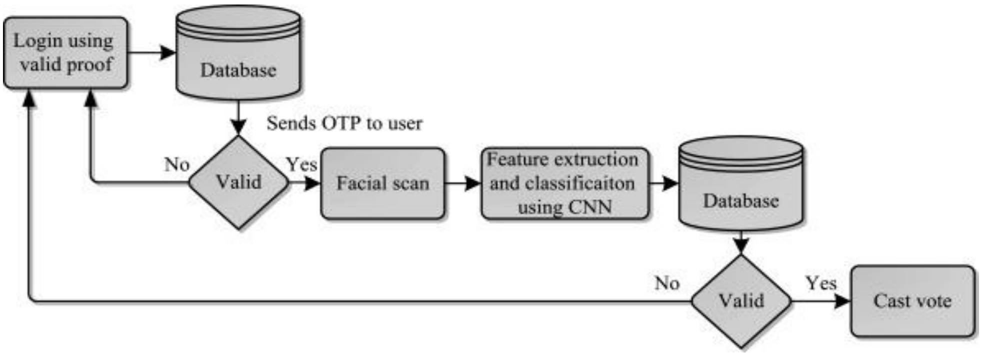


Figure 2: CNN Flow chart

Flowchart Steps Explained

1) **Start**: First, we need to input an image. This could be a 2D grid of pixel values for a colorful image with RGB channels.
2) **Input Image:** This image is the data that our CNN (that's short for Convolutional Neural Network) uses.
3) **Convolution Layer:** Next up, we apply filters. These help us to find patterns in the input image.
4) **ReLU Activation:** We use something called the ReLU activation function. It helps add some non-linearity to the process.
5) **Pooling Layer:** Now, we move on to the pooling layer. Here, we reduce the size of the feature maps by using pooling techniques like max pooling.
6) **Repeat Convolution and Pooling Layers:**We can repeat those convolution and pooling steps multiple times. Each time we add more layers. This helps us find more complex features.
7) **Flattening:** After that, we flatten everything. This means turning our 2D feature maps into a long 1D vector so we can feed it into fully connected layers.
8) **Fully Connected Layer:** In this step, we do classification. The data goes through these fully connected layers (also known as dense layers).
9) **Output Layer:** Finally, there's the output layer. It usually uses a softmax or sigmoid function to give us probabilities for each class.
10) **Final Output:** Then comes the fun part the class label that's predicted by the CNN.
11) **End:** We end with the final result. For example, it might say the image is classified as a dog or cat, etc.

3.7 PROPOSED SYSTEM

The system being proposed includes a secure method for voting using facial recognition technology. This system utilizes machine learning to authenticate voters by analyzing their faces and matching them with images in a database of registered voters. Once the verification is successful, users can securely submit their votes through an online voting platform. This ensures that all votes are encrypted and stored securely to uphold the integrity of the electoral process.

- ✓ **Enhanced Security:** Using face recognition makes voting safer, lowering the chance of someone pretending to be another voter. It's harder to fake biometric info than regular IDs.
- ✓ **Convenience:** Voters can vote from anywhere they want No need to go to polling places helps those who find it tough to get there.
- ✓ **Reduced Fraud:** With this system, one voter can only vote once, making sure everyone gets just one voice in decisions.
- ✓ **Efficiency and Speed:** This method speeds up checking votes and counting them. Plus, it cuts down on mistakes we see with manual checks.

The Online Voting System using Face Recognition is designed to make elections safe, simple, & super easy for everyone. Here are the main parts and features of the system:

User Registration: To join users fill out a form with their name, email, mobile number, & Aadhaar number. They also need to upload 20 pictures of their face. These pictures help train the face recognition system. This way, it can recognize the user when they log in.

This part uses smart tech like machine learning to get everything just right using tools like OpenCV along with models like convolutional neural networks (CNN).

Face Recognition for Authentication: After registering, individuals can access the system by capturing an image of their face using their device's camera. The system utilizes advanced algorithms such as CNN, Haar Cascades, or other deep learning techniques to verify if their face corresponds to any of the 20 images they previously uploaded. In addition to this, users will be required to input their email and password. For added security, they will use a one-time password (OTP) to prevent any unauthorized access. If all the details align and the OTP is accurate, they will be granted immediate access to the system.

Voting Process: After logging in, users can directly access the voting section to select candidates or issues in the election. A user-friendly interface guides voters through each step, making the process easy for everyone, including those who aren't very tech-savvy. The system ensures that each voter can only cast their vote once, preventing any duplicate or fraudulent votes from occurring.

Security Features:

- ✓ **Face Recognition Model:** A strong model keeps face matching safe. It prevents fraud like someone trying to vote for another person.
- ✓ **Data Encryption:** All personal info and facial images are covered with strong encryption in the database.
- ✓ **Block chain for Voting Integrity:** For extra safety on votes cast, the system uses blockchain technology. That means once a vote is cast, it can't be changed or messed with.

Admin Dashboard: Admins have control over the whole voting system. They can add candidates, set election dates, and check results easily. They also get to see detailed logs of who logs in and who votes. This helps keep everything running smoothly & safely.

System Benefits:Enhancing secure voting can be achieved through the use of face recognition technology, which verifies the eligibility of individuals before allowing them to cast their vote. By automating the verification process, time efficiency is improved, eliminating the necessity for passwords and the inconvenience of resetting them. Furthermore, it aids in preventing fraud by ensuring that each voter only casts one valid vote after a confirmed facial match.

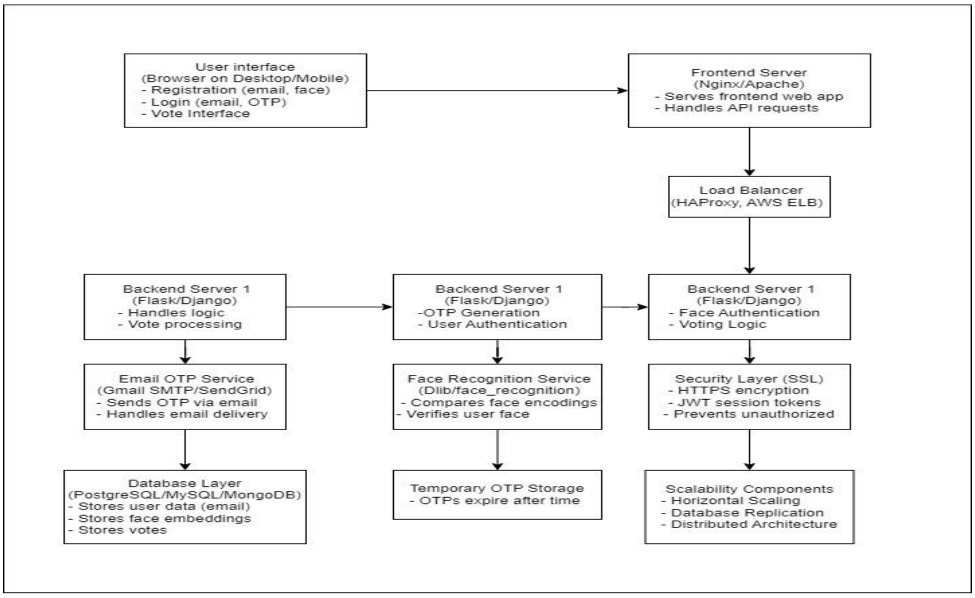


Figure 3: System Architecture

Expalination:

1. **User Interface:** This is the front-end of the system, accessible to users through a browser on desktop or mobile devices. It allows users to register, log in, and cast their votes.
2. **Frontend Server:** This server handles the presentation and interaction of the user interface. It serves the web app and processes API requests from the user interface.

3. **Load Balancer:** This component distributes incoming traffic across multiple backend servers to improve performance and scalability.
4. **Backend Servers:** These servers handle the business logic of the system, such as user authentication, OTP generation, face authentication, vote processing, and data storage.
5. **Email OTP Service:** This service sends OTPs (One-Time Passwords) to users via email for verification purposes.
6. **Face Recognition Service:** This service compares face encodings to verify user identity.
7. **Security Layer:** This layer ensures the security of the system by using HTTPS encryption and JWT session tokens to prevent unauthorized access.
8. **Database Layer:** This layer stores user data, face embeddings, and votes.
9. **Temporary OTP Storage:** This component stores OTPs for a limited time to prevent unauthorized access.
10. **Scalability Components:** These components enable the system to handle increased load by allowing for horizontal scaling and database replication.
11. **Distributed Architecture:** This architecture distributes the workload across multiple servers to improve performance and reliability.

4. RESULTS:

4.1 Objective:

Let's create a simple and safe online voting system that uses face recognition to help users log in. This system makes it easy to sign in with a face instead of using passwords. It's all about making things safer and more convenient.Components used are as follows:

User Registration:

Information Collected:

Name

Email Address

Mobile Number

Aadhar Number

Image Collection: Users need to send in 20 pictures of themselves. This helps train the facial recognition model.

Image Storage: All images are kept safe in the database for training the face recognition model.

4.2 Login Process:

Facial Recognition: When logging in, users take a picture with their camera. The system checks this picture against stored images.

Three-Step Login: If the facial recognition finds the user and if the OTP & Password are correct, they get logged in right away.

4.3 Technologies Used:

Face Recognition Library: We use OpenCV, dlib, or face_recognition for spotting faces.

Backend Framework: Flask, Django, or Node.js helps manage user logins & data handling.

Database: MySQL or PostgreSQL saves user info and pictures.

Frontend: HTML, CSS & JavaScript show a friendly interface for users.

Camera Integration: Having hardware or software solutions for capturing images.

4.4 User Experience:

Simplified Login: Users log in fast with facial recognition & get a quick OTP to their email.

Convenience: Logging in is smooth and modern, which makes it better overall.

4.5 Security:

Enhanced Security: Facial recognition is more secure than just using passwords.

Authentication Accuracy: It relies on how well the face recognition can spot faces. It has to be trained with good-quality images.

4.6 System Performance:

Speed: The system must recognize faces quickly for a nice login experience.

Scalability: It should handle many users logging in at busy times without a hitch.

4.7 Challenges:

Image Quality: Pictures used must be good; not clear ones can mess up logins.

Privacy Concerns: Storing personal data means following privacy rules so no one misuses information.

4.8 Future Enhancements:

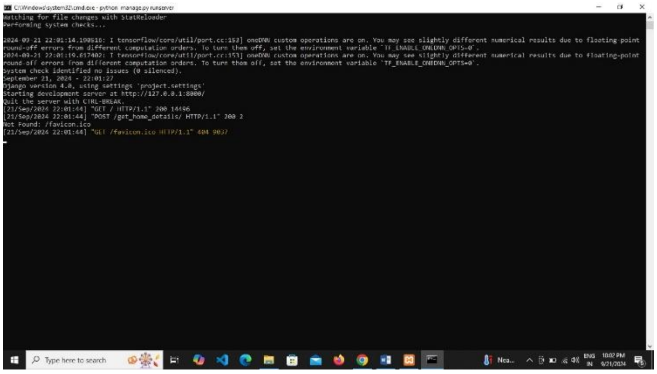
Continuous Learning: Using smart algorithms that keep learning to spot faces better when new data comes in

Multi-Factor Authentication: Mixing facial checks with other verification methods can boost security even more.

4.9 User Feedback:

Collecting what users think helps make the system even better and fixes any problems.

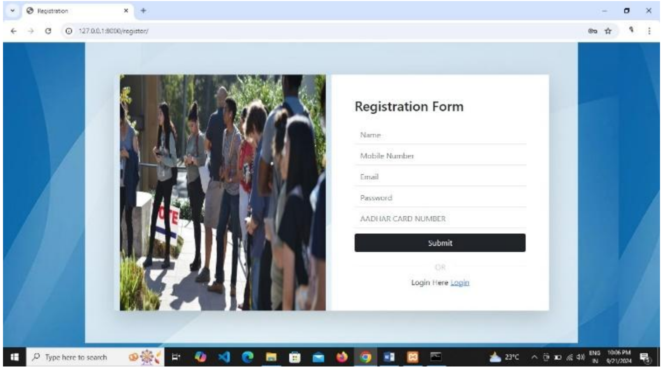
SCREENSHOTS:



```
C:\Windows\system32\cmd.exe - python  manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

2024-09-21 22:01:14.198516: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point
round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-09-21 22:01:19.617402: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point
round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
System check identified no issues (0 silenced).
September 21, 2024 - 22:01:27
Django version 4.0, using settings 'project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[21/Sep/2024 22:01:44] "GET / HTTP/1.1" 200 14496
[21/Sep/2024 22:01:44] "POST /get_home_details/ HTTP/1.1" 200 2
Not Found: /favicon.ico
[21/Sep/2024 22:01:44] "GET /favicon.ico HTTP/1.1" 404 9037
```

Figure 4: Program running



127.0.0.1:8000/register/

Registration Form

Name

Mobile Number

Email

Password

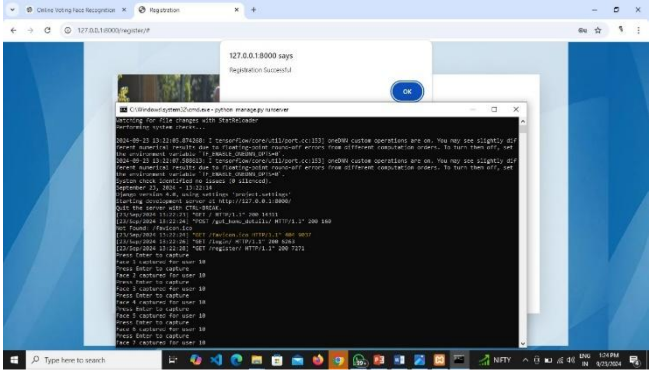
AADHAR CARD NUMBER

Submit

OR

Login Here Login

Figure 5:Registration Details



127.0.0.1:8000/register/#

127.0.0.1:8000 says

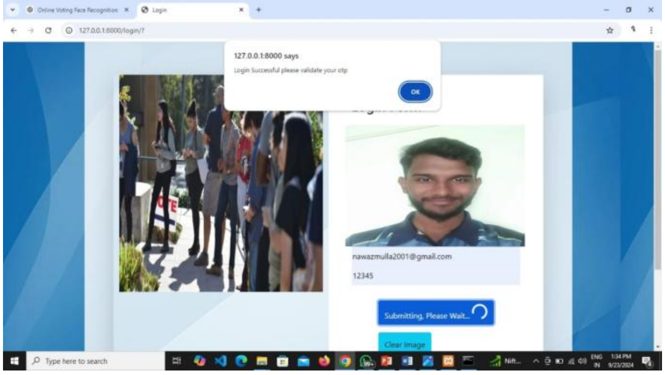
Registration Successful

OK

```
C:\Windows\system32\cmd.exe - python  manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

2024-09-23 13:22:05.874268: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly dif
ferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, set
the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-09-23 13:22:07.588613: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly dif
ferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, set
the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
System check identified no issues (0 silenced).
September 23, 2024 - 13:22:14
Django version 4.0, using settings 'project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[23/Sep/2024 13:22:23] "GET / HTTP/1.1" 200 14311
[23/Sep/2024 13:22:24] "POST /get_home_details/ HTTP/1.1" 200 168
Not Found: /favicon.ico
[23/Sep/2024 13:22:24] "GET /favicon.ico HTTP/1.1" 404 9037
[23/Sep/2024 13:22:26] "GET /login/ HTTP/1.1" 200 6263
[23/Sep/2024 13:22:28] "GET /register/ HTTP/1.1" 200 7271
Press Enter to capture
Face 1 captured for user 10
Press Enter to capture
Face 2 captured for user 10
Press Enter to capture
Face 3 captured for user 10
Press Enter to capture
Face 4 captured for user 10
Press Enter to capture
Face 5 captured for user 10
Press Enter to capture
Face 6 captured for user 10
Press Enter to capture
Face 7 captured for user 10
```

Figure 6: Image Capturing



127.0.0.1:8000/login/?

127.0.0.1:8000 says

Login Successful please validate your otp

OK

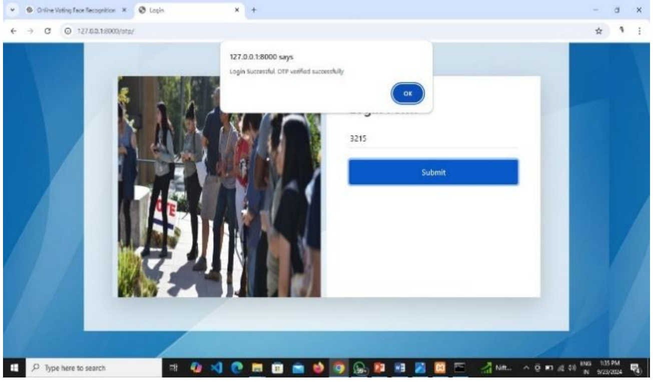
nawazmulla2001@gmail.com

12345

Submitting, Please Wait...

Clear Image

Figure 7:Login with Face Data & OTP



127.0.0.1:8000/otp/

127.0.0.1:8000 says

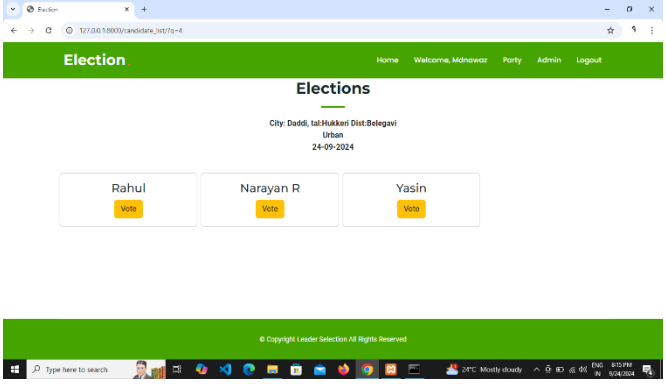
Login Successful. OTP verified successfully

OK

3215

Submit

Figure 8: OTP Verification



127.0.0.1:8000/candidate_list/?q=4

Elections

City: Daddi, tal:Hukkeri Dist:Belegavi
Urban
24-09-2024

Rahul Vote

Narayan R Vote

Yasin Vote

© Copyright Leader Selection All Rights Reserved

Figure 9:Voting Page.

Expalination:

As shown in the (*Figure 4)* we need internet connection to host the Program in the web browser. It is done by giving some specific commands to run the Porgram.Next step is Registration of the User or Voter we can say. User needs to provide some personal informtions as shown in the (*Figure 5)*. After this the user will be directed to command prompt for collecting thr face data. The system captures 20 images for each registration as shown in the (*Figure 6*) During the login process the user needs to provide their face data along with Email and password which is shown in the (*Figure 7)* if the data matches then the user will receive an Pop up and will be directed to the OTP verification Page .An OTP will be sent to the User's Registered mail user need to verify the otp if it verifies then the user will receive the Pop up as shown in the (*Figure 8)*. Then the user will be directed to the user Home page where the Election Details will be displayed according to the dates after clicking the Article an UI will be displayed in which the candidates along with their Parties from where the Voters can elect their Leaders By voting them as shown in the (*Figure 9).*

5. CONCLUSION

An online voting system that utilizes facial recognition a path toward improving election security, accessibility, and convenience. Yet, to unlock its full capabilities, future efforts must tackle issues concerning accuracy, privacy, bias, and the ability to scale. Incorporating cutting-edge technologies such as liveness detection, privacy-enhancing techniques, and blockchain can enhance the system's dependability. It's also vital to ensure inclusivity and adherence to legal standards to build public confidence. With ongoing research and innovation, this strategy has the potential to transform the voting process, making it more secure, transparent, and available to everyone.

FUTURE WORK:

Futuredevelopments for the online voting system utilizing facial recognition should prioritize enhancements in accuracy, security, and accessibility. A significant focus should be on refining facial recognition algorithms to minimize bias across various demographic groups and to boost in difficult situations, such as low-light environments. Implementing advanced anti-spoofing methods that leverage liveness detection can help thwart fraudulent activities involving photos, videos, or deep fake technology.

ACKNOWLEDGEMENT:

I am grateful to Prof.ShivanandGornaleDepartment of Computer Science, Rani Channamma University, Belagavi for his valuable guidance for completion of this work

REFERENCES:

[**1**] M. D. Rodriguez and G. L. Marcialis, "Face recognition applications and challenges in online voting systems," in *Proceedings of the International Conference on Biometrics*, Seoul, South Korea, 2018, pp. 203-208.

[**2**] M. Gamassi et al., "Improving voter authentication through facial and fingerprint biometrics: A comparative study," *Journal of Electronic Voting*, vol. 15, pp. 203-208, 2020.

[**3**] S. Marcel and C. McCool, "An analysis of face recognition in the context of online voting," *International Journal of Computer Vision*, vol. 104, pp. 287-298, 2022.

[**4**] H. Dibeklioglu et al., "Robust face recognition using liveness detection in voting systems," *Pattern Recognition Letters*, vol. 56, pp. 123-131, 2022. https://doi.org/10.1016/j.patrec.2022.05.003

[**5**] C. Rathgeb and A. Uhl, "A survey on biometric face recognition for online voting systems," *IEEE Access*, vol. 7, pp. 8977-8985, 2022. https://doi.org/10.1109/ACCESS.2022.3145235

[**6**] F. Ferri et al., "GDPR compliance in online voting systems using face recognition," *Journal of Digital Security and Privacy*, vol. 8, pp. 43-57, 2022. https://doi.org/10.1016/j.jdsp.2022.05.002

[**7**] N. Srivastava et al., "Liveness detection in facial recognition for secure voting systems," *IEEE Security and Privacy*, vol. 15, no. 4, pp. 49-58, 2022. https://doi.org/10.1109/MSP.2022.3001348

[**8**] A. Ross, "Hardware requirements and their impact on biometric voting systems," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 3, pp. 215-225, 2021. https://doi.org/10.1109/TBIOM.2021.3058745

[**9**] M. Gamassi et al., "Improving voter authentication through facial and fingerprint biometrics: A comparative study," *Journal of Electronic Voting*, vol. 15, pp. 203-208, 2020. https://doi.org/10.1016/j.jev.2020.05.009

[**10**] A. K. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270-2285, 2019. https://doi.org/10.1016/j.patcog.2019.01.005

[**11**] S. Marcel and C. McCool, "An analysis of face recognition in the context of online voting," *International Journal of Computer Vision*, vol. 104, pp. 287-298, 2022. https://doi.org/10.1007/s11263-022-01525-4

[**12**] F. Ferri et al., "GDPR compliance in online voting systems using face recognition," *Journal of Digital Security and Privacy*, vol. 8, pp. 43-57, 2022. https://doi.org/10.1016/j.jdsp.2022.05.002

[**13**] N. Srivastava et al., "Liveness detection in facial recognition for secure voting systems," *IEEE Security and Privacy*, vol. 15, no. 4, pp. 49-58, 2022. https://doi.org/10.1109/MSP.2022.3001348.