Mobile Application Testing with AI Features (Envision AI)

Atharva Munshi, Noopur Mehta, Mitra Gunakara Nayak, Bhumika Tiwari

Abstract—Rapidly changing technologies along with Artificial intelligence is having a huge impact on society. Image recognition has been the basis of almost every AI innovation. One such application which makes use of image processing and recognition is Envision AI. This App basically helps visually impaired users by enabling them to shop in supermarkets, recognize their friends, etc. In this project, We will test the Envision AI application, which will for the most part center around testing the AI part of the Application.

Index Terms—AI models, Testing, AI Testing, Manual testing

1 Introduction

THE The way technology has changed our lives has become quite evident from the last couple of decades and it continues to grow with each passing day. Amid this,Artificial Intelligence(AI) has become one of the most rapidly accepted and used technologies all around the globe. Undoubtedly, SIRIs and Alexa have made our lives easy. However, use of AI is not limited to just that. If we look around, we can see a lot of applications that implement and use Image recognition for AI Advancement. On such application is the Envision AI application. Envision Helps blind people and makes their everyday life easier by helping them carry out the tasks like reading newspapers to identify people.Image Recognition is the ability of a machine or a software to recognize a particular thing or an activity. In this project, We will test the Envision AI application, which will for the most part center around testing the AI part of the Application. In this project, we investigated the applicability of Twitter data in the realm of politics. This is especially relevant at this time since the US presidential primaries are ongoing, as a result of which millions of people are actively expressing their views about each candidate. We also compared our results with public opinion polls. We have provided detailed analysis using data visualization to quickly understand how public sentiment about each candidate evolved over time and place. This is further broken down by the key issues like healthcare and economy.

2 Envision AI Features

- Human/Object Detection
- Text Detection
- Color Detection
- Barcode Scanning

3 Types of Testing

3.1 Conventional(/Manual) Testing

One of the core parts of SDLC is the test models. In order to develop a software, different models are used. There are several pros and cons of each model. Thus. If we should select a particular model or not completely depends on the type of project as well as the given requirements of that project. After going through different models, we decided to use the waterfall model for our project. The reason this model benefits our project is that there is no connection and dependency in the development process. Apart from that, we have decided to perform the testing of our project independently. The phases of the model are:

- Requirement gathering and analysis
- Design the Tests (System design)
- Implementation
- Execute the Tests
- Deployment
- Maintenance

The testing which will be done by taking this model into consideration will be a linear process. There will be a minimum amount of feedback loop. This loop is generated because the problems found because of the test failures should be fixed. The pros and cons of using this model are as follows: **Pros:**

- It is not complicated and is easily understandable.
- Less time is required for the later phases if the initial phase is performed properly.
- Testing is performed at the end of every phase.

Cons:

- The requirements are not changeable, and we cannot go back to previous phase in order to make the changes
- In order to start the next phase, the current phase should be completed.

We have decided to use black box testing. In black box testing, the internal working of the program is not taken into consideration. More weight-age is given to inputs and outputs. The functions of the code are tested. The knowledge of how the implementation is done or how the code is

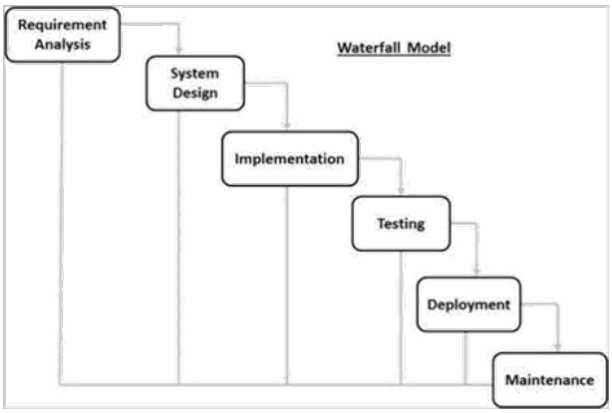

Fig. 1. Waterfall model

internally wired is not required. There are different types of black box testing:

- Functional testing
- Non-functional testing
- Regression Testing

As the name suggests, functional testing tests the functions of the program. In case of Non-functional testing, testing is done on the maintainability, usability, scalability, etc. Regression testing is done when we need to check the maintenance of the code. There are different methods of black box testing:

1) Category Partition Testing
2) State Transition Testing
3) Decision Table Testing
4) Boundary Value Testing
5) Equivalence Partitioning Testing
6) Scenario Testing

We can note the test coverage for all of the above methods and know the coverage of the tests.

3.2 AI Testing

3.2.1 Artificial Intelligence Usage and Test Requirement - Envision AI

The Envision AI is making use of unsupervised Machine learning, image recognition, machine vision, text generation, text to speech AI functionalities. Our test requirement revolves around testing each of the above features and determining the AI efficiency of the envision AI application. To achieve the same, we would be using different contexts, inputs and considering the possible outputs to arrive at the accuracy and consistency of the application. Basically our test requirement would be testing each of the function mentioned below

1) Color detection
2) Object Identification
3) Text Identification
4) Barcode Scan
5) Human/Face Detection

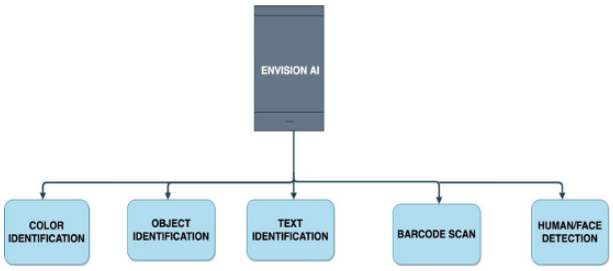

Fig. 2. AI Features

3.2.2 AI Test Modeling for your selected AI Features

Test Modeling for AI features comprises of several kind of modeling activities which are as follows:

1) Context Modeling
2) AI Function Input classification
3) AI Function Output classification
4) AI Function Event/Action Classifications

Thus, for testing the AI features, context needs to be identified first. Then, the input data set for the AI needs to be listed. Under the context modeled and AI inputs classified, the list of output generated needs to be identified and classified as output sets. This conversion from input to output would happen as a result of events or actions. These actions can also be classified as action/event sets. Below, we have performed test modeling for AI features for Envision application.

Context Modeling for AI Features:

For testing of AI features in the Envision app, context modeling has been done to arrive at the spanning tree shown in the figure below. Context is majorly categorised on the basis of below factors:

- Environment in which the Envision app will be tested. Environment can further be classified into indoor and outdoor test conditions.
- Position of the device when the camera is pointed on an object in focus. It can be in an acute angle or obtuse angle or in a parallel position with respect to the object.
- Distance between device and object when camera is pointed on an object in focus. It can be extremely close or extremely far or at optimal distance from the object.
- Lighting conditions in the environment in which the object under focus exists. There can be no light or dim light or optimal light or bright light environment.
- Stability of the object in focus. It can be a still object or one in motion.
- Equipment type on which Envision app is installed and to be used for testing. Android and iOS variants are the subcategories in this context.

AI Function Input Classifications

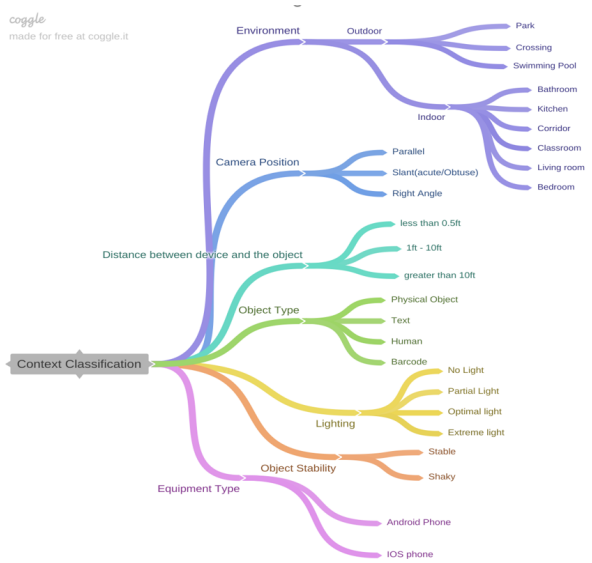

Fig. 3. Context Diagram

Fig. 4. Input Classification

Fig. 5. Output Classification

The input dataset to Envision app can be classified into categories as shown in the diagram below. Input to Envision app classification is necessary to identify all possible functional scenarios for testing of AI features:

- Object: Envision app camera can be directed at single or multiple objects when testing for objects in a scene.
- Human: Envision app camera can be directed at a human of either male or female gender for testing. The person in focus can be involved in any activity like reading, playing, walking.
- Text: Text in focus of Envision app can be either handwritten text or printed text. Text font size can be too large or too small or optimal. Text language under test is Hindi, English and numerical data. Text content can be a letter or a word or an entire paragraph.
- Color: The object in focus of Envision app can have a primary color or secondary color or tertiary color.
- Barcode: The barcode to be scanned by the Envision app can be present on a physical object or display screen of some device like a computer or tablet.

AI Function Output Classifications The output from Envision app can be understood as a set which can further be categorised as shown in the figure below.

The output of Envision app AI features testing can be segregated into below types which can further be divided into different sets of observations:

1) Color:
   - The Envision app can identify the color of an object correctly.
   - The Envision app cannot identify the color of an object correctly.
2) Object:
   - Multiple objects in focus of Envision app -
     a) The Envision app can identify multiple objects in focus correctly.
     b) The Envision app cannot identify multiple objects in focus correctly.
   - Single objects in focus of Envision app -
     a) The Envision app can identify single objects in focus correctly.
     b) The Envision app cannot identify a single object in focus correctly.
3) Human:

- Person in focus of Envision app of either gender

  a) The Envision app can identify the correct gender of a person in focus.
  b) Envision apps cannot identify the correct gender of a person in focus.

- Person in focus of Envision app performing some action -

  a) The Envision app can identify the action being performed by a person in focus.
  b) Envision app cannot identify the action being performed by a person in focus.

4) Barcode:

- The Envision app can scan the barcode of an object correctly and determine the object to which the barcode actually belongs.
- The Envision app is unable to scan the barcode of an object correctly.

5) Text:

- Type of text, i.e. handwritten or typed - The Envision app can read a piece of handwritten text.
- The Envision app can read a piece of typed text.
- The Envision app cannot read a piece of handwritten text.
- The Envision app cannot read a piece of typed text.

  – Text of different languages, Hindi, English and numeric data tested - The Envision app can interpret language text of different languages correctly.
  – The Envision app cannot interpret language text of different languages correctly.

3.2.3 AI Function Event/Action Classifications

Envision AI application can perform several AI functions each of which is triggered by an event in time. Depending on what functionality is selected by the user, the app takes visual input from the camera and processes it to provide the expected output.

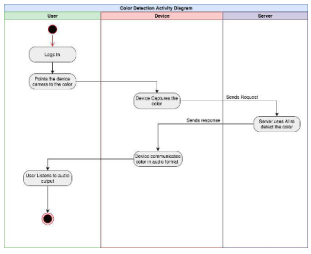

Fig. 6. Color Detection

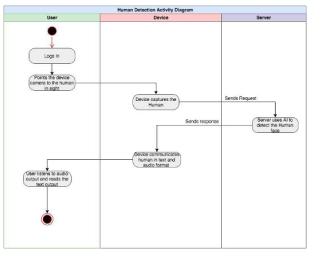

Fig. 7. Human Identification

3.2.4 AI Function Classification Decision Table(3D tables)

The below decision tables represent the 3D view of each of the AI functionality.
The 3 faces of the 3D model are

1) AI function context classification view
2) AI function input classification view
3) AI function output classification view

Please find 3D diagrams for all four AI features of Envision AI Mobile Application.
Input and Output classifications are designed keeping in mind the Context diagram and We have tried to add all the possible factors/ contexts.

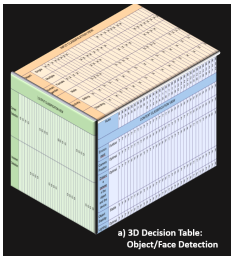

Fig. 8. 3D diagram for Object/Face detection

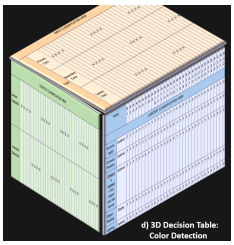

Fig. 9. 3D diagram for Color detection

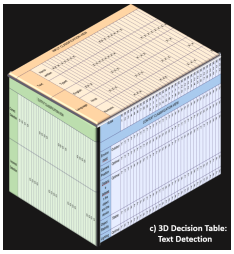

Fig. 10. 3D diagram for Text detection

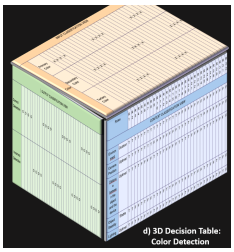

Fig. 11. 3D diagram for Barcode detection

3.3 AI Function Test Cases with Inputs/Expected Outputs

There are a lot of methods used for testing AI functionalities of Software Applications and/or websites depending on their respective functionalities and Test requirements. **Test data models AI Software Testing Models:**

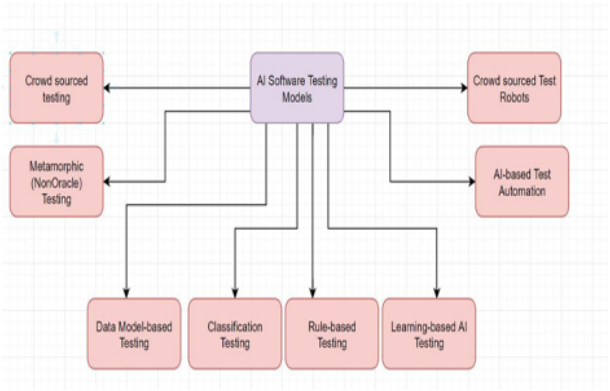

Fig. 12.

Crowdsourced Testing: Crowdsourced Testing refers to usability testing by a bunch of different Testers from different locations. It is a user centric testing method that focuses on Testing from User perspective, emphasises the most on User feedback. In addition to an inhouse Quality Assurance/ Testing team, the application/ Software is outsourced to be tested by different users to understand the System from their point of view. Eg. User testing website (https://www.usertesting.com/) : A lot of technology giants put their Software Product to be tested on this website and seek user feedback by paying for the video feedback.
Metamorphic(Non-Oracle) Testing

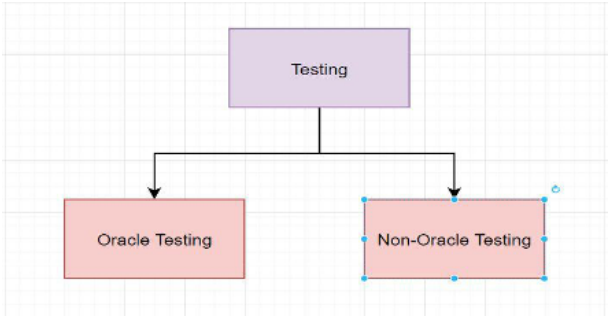

Fig. 13.

Oracle Testing: Oracle Testing is a manual testing method where we have fixed input and fixed expected output. Disadvantages of Oracle Testing:

1) Expensive to design manual test cases
2) Cannot guarantee the coverage of all the Test cases.

Non-Oracle Testing: Non-oracle testing(Metamorphic Testing) is a Property Based Testing(PBT) where the inputs and expected outputs are not fixed. This technique is used hugely especially for AI Testing since the requirements and constraints are more likely to change with time in the same. Metamorphic Relations:It is a mathematical property that defines relations between the given input and outputs.So, instead of validating output, validate the relation between input and output. The main benefit of using this method is that we can expect the output to change in the same way input does. If that doesn't happen, Test case is marked as Fail. Problem:It is complex to generate uniform code. Solution: Convert the code into Context Flow Graph(CFG), Train a model and finally determine the metamorphic relation.
Data Model-based Testing In Data model- based testing,Runtime behavior of a Software under test is checked against predictions made by any of the below described models.

1) Data flow diagram
2) Control flow diagram
3) Dependency graphs
4) Decision tables
5) State Transition Machines

Classification Testing Classification Testing is a method of AI-Testing that closely resembles one of the Black-box testing methods, Category Partition Testing. We have to create context trees for each AI- Functionality for both inputs and Possible outcomes and design the test Cases keeping in mind all the possible combinations.
Rule-based Testing As the name suggests, Rule-based technique involves Hard-coded rules, that cannot be changed in the future with the requirements. AI Implemented through rule-based technique has a Fixed amount of knowledge and limited scope. Disadvantages:

- Rules cannot be changed. Hence, it is time consuming and expensive.
- In some situations, It is not possible to define rules explicitly.

Learning-based AI Testing

- As the name suggests, it is general AI with the Learning Capabilities.
- Artificial Intelligence in this method is implemented and tested using Machine Learning Techniques.
- No hard rules, you can build rules on the fly.Ex. Neural Network

AI-based Test Automation For automatic Test Case generation, especially for Soft wares/ Applications having AI features, there are several tools available online. Following are the AI Test Automation Tools:

- Applitools
- SauceLabs
- Testim

- Sealights
- Test.AI
- Mabl
- ReTest

Crowdsourced Test Robots Crowdsourced Test Robots are the third party service providers that facilitates AI Automation Testing with a few simple modifications and automatically generates Test reports as well as bug reports.
Disadvantages:

- Can be ambiguous if the algorithm is not trained properly
- Expensive

4 Automation Testing

The purpose of this project is to perform AI testing for Envision mobile applications. We performed conventional testing for the app earlier and this document is intended to provide the details for automation testing for the AI functions of this app using Appium. Although conventional test execution did uncover some defects in the application under test, yet test automation was performed with the intent of uncovering more bugs.

4.1 Automation Testing Objectives

- Focus on engineers to save manual efforts for performing repetitive tasks.
- Focus on accelerating the pace of work.
- Focus on reducing cost.
- Focus on achieving better quality software products.

4.2 AI Test Scenarios

We have designed AI Test scenarios from input and output classification for writing Test Scripts, some of which are as follows.

1) Human and Object Detection
   - Man riding a bicycle.
   - Laptop placed on a table.
2) Text Identification
   - Text present on a computer screen.
   - Handwritten text present on a sheet of paper.
3) Color Detection
   - Color present on physical object.
   - Color present on a computer screen.
4) Barcode Scanning
   - Barcode present on physical object.
   - Barcode present on a computer screen.

4.3 Sample Appium Test Script

We have developed automated test scripts for each AI functionalities of the system. Please find a sample test script below.

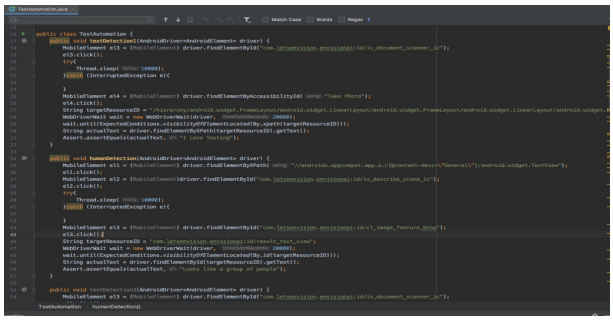

Fig. 14. Automation script snippet

5 AI Automation Testing Results

5.1 Test Case Results for Text Identification

From the chosen Scenarios, we performed automated testing on the Envision AI application, executed the test cases and got the below mentioned results. Scenario: ”I Love Testing” text present on a Computer Screen. application, We have written test scripts considering different context such as lighting condition, camera angle, Distance between camera and object, Language of text,etc.

Test Case#	Test Case	Expected Result	Actual Result	Test Execution Outcome
2.2.1.1	Light - optimal Camera angle - optimal Distance - optimal Color of Text - black Size of Text - optimal Language of text- English	The App should read "I love Testing"	The App reads "I love testing "	Fail
2.2.1.2	Light - **Dark** Camera angle - optimal Distance - optimal Color of Text - black Size of Text - optimal Language of text- English	The App should read "I love Testing"	The App reads "No Text Found"	Fail
2.2.1.3	Light - **Too Bright** Camera angle - optimal Distance - optimal Color of Text - black Size of Text - optimal Language of text- English	The App should read "I love Testing".	The App reads "No Text Found"	Fail
2.2.1.4	Light - optimal Camera angle - **Slant** Distance - optimal Color of Text - black Size of Text - optimal Language of text- English	The App should read "I love Testing"	The App reads "I love testing "	Fail
2.2.1.5	Light - optimal Camera angle - optimal Distance - **Too close** Color of Text - black Size of Text - optimal Language of text- English	The App should read "I love Testing"	The App reads "I love testing "	Fail

Fig. 15. Test Case Results

5.2 Test Statistics

Below you can find Test Result statistic for human/ object detection feature of Envision AI.

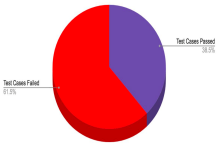

Fig. 16. Human/object detection

Test Case Matrix	Total number of Test Cases	Test Cases Passed	Test Cases Failed
Count	13	5	8
Percentage	100%	38.5%	61.5%

Fig. 17. Human/object detection

- For testing Envision mobile application which offers multiple AI features, we have used classification

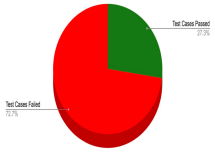

Fig. 18. Color detection

Test Case Matrix	Total number of Test Cases	Test Cases Passed	Test Cases Failed
Count	22	6	16
Percentage	100%	27.3%	72.7%

Fig. 19. Color detection

based test strategy. As part of this strategy, we performed classification modeling for contexts, inputs and outputs.

- When input is fed to the app under test, events are triggered under a defined context environment to generate output. This test strategy helps in achieving maximum test coverage under diverse test conditions and inputs.
- To leverage the advantages of classification based strategy, 3D Decision tables have been used in testing of this project. The three views of the decision cube AI function context classification view, AI function input classification view, AI function output classification view ensured better test coverage of Envision AI application.
- We have used Spanning Trees extensively during test modeling for context, inputs, outputs and events. To test the project, we have used a real device testing approach wherein the mobile device with Envision app installed was connected to a computer on which Appium server was installed and running. We have written Java test scripts for testing the AI functions.
- The main focus of Automated Testing is to provide test coverage of a cumulative nature, uncover defects, lower failure costs and facilitate repetitive tasks to run in an automated manner. This helps in keeping software cost to market under control and optimize usability of resources.

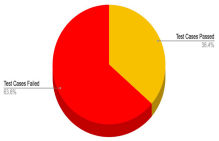

Fig. 20. Text detection

Test Case Matrix	Total number of Test Cases	Test Cases Passed	Test Cases Failed
Count	22	8	14
Percentage	100%	36.4%	63.6%

Fig. 21. Text detection

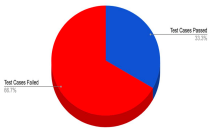

Fig. 22. Barcode detection

Test Case Matrix	Total number of Test Cases	Test Cases Passed	Test Cases Failed
Count	12	4	8
Percentage	100%	33.3%	66.7%

Fig. 23. Barcode detection

5.3 Comparison: Conventional AI Test Results vs Automated AI Test Results

From the chosen Scenarios, we performed automated testing on the Envision AI application, executed the test cases and got the below mentioned results.

Manual Testing is done manually by QA analyst (Human) whereas Automation Testing is done with the use of script, code and automation tools (computer) by a tester.

Manual Testing process is not accurate because of the possibilities of human errors whereas the Automation process is reliable because it is code and

script based. Manual Testing is a time-consuming process whereas Automation Testing is very fast.

Manual Testing is possible without programming knowledge whereas Automation Testing is not possible without programming knowledge.

Manual Testing allows random Testing whereas Automation Testing doesn't allow random Testing.

The manual testing process can't be recorded, so it is not possible to reuse the manual test.

Automated Testing is suited for Regression Testing, Performance Testing, Load Testing or highly repeatable functional test cases.

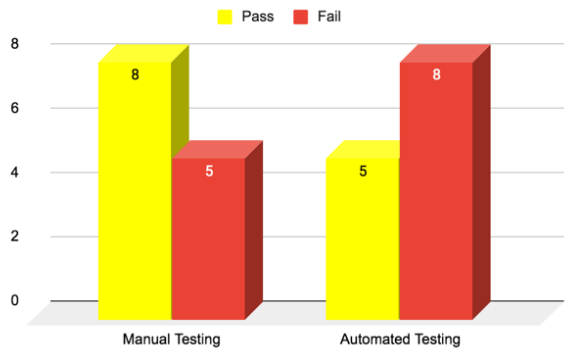

Fig. 24. Comparison

Manual vs Automated	Total number of Test Cases	Pass	Fail
Manual Testing	13	8	5
Automated Testing	13	5	8

Fig. 25. Comparison

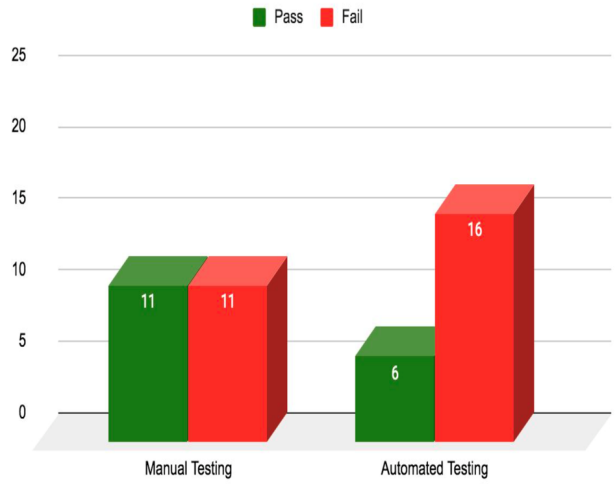

Fig. 26.

Manual vs Automated	Total number of Test Cases	Pass	Fail
Manual Testing	22	11	11
Automated Testing	22	6	16

Fig. 27. Comparison

- As the market's dependency on technology grew, companies needed additional features to be shipped faster, to keep pace with different user needs and to stay ahead of the competition. Rapid development and adoption of Agile methodologies moved from being nice to have to must have.
- Faster development cycles were introduced as a part of Agile methodologies that aimed to implement new features within a sprint that lasted a few weeks. However, while keeping the development cycles shorter, the time dedicated to software testing was reduced. This resulted in more bugs and a bad experience for users on less popular devices, who formed a long tail.

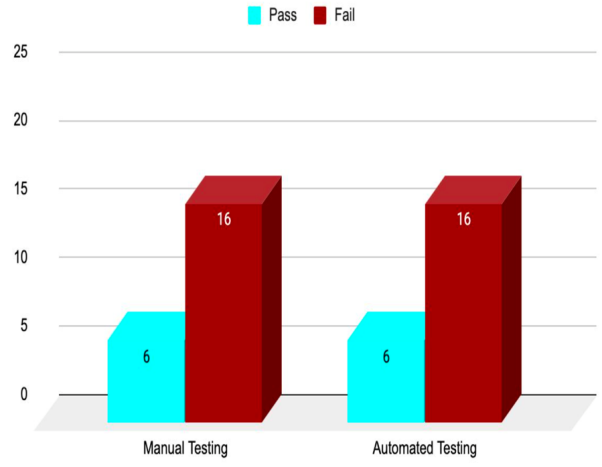

Fig. 28.

Manual vs Automated	Total number of Test Cases	Pass	Fail
Manual Testing	22	6	16
Automated Testing	22	6	16

Fig. 29. Comparison

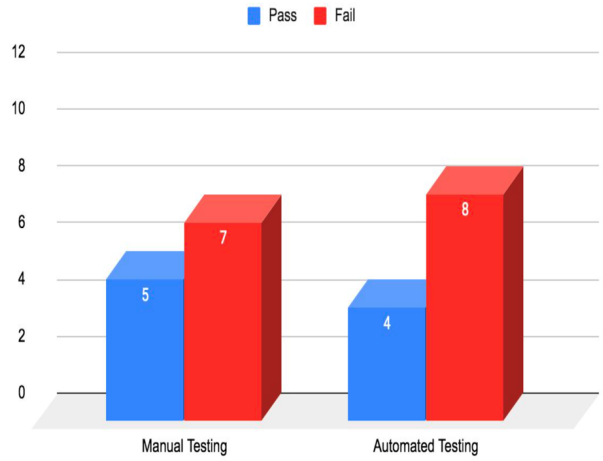

Fig. 30.

- Although Manual testing performs well in areas that require quick early results and analysis, it does not yield fruitful results for testing areas, that require repeated iterations and execution of the code. It also cannot match up when the scale is huge, as it consumes time, and could end up causing unnecessary delays in a fast-moving technology space.
- This is where Automation Testing comes into the picture. The ability to run iterative, parallel tests on multiple devices, browser versions and operating systems in one go, with error logs and reports automatically generated, can easily be the difference between a market leader and a weak competitor.
- Keeping in mind the need for maximum device coverage, short timeframe of the sprints, and the need to keep costs low, Automation selenium testing on cloud seemed to be the best option.
- Automation, no matter how smart it is, still lacks the cognitive abilities and cannot depict human-like intelligence when it comes to decision-making.

Manual vs Automated	Total number of Test Cases	Pass	Fail
Manual Testing	12	5	7
Automated Testing	12	4	8

Fig. 31. Comparison

5.4 Defects

Please find the sample of defects found in each AI functionality of Envision AI application.

Defect Description	Severity
The application does not detect human and objects in focus together.	Major

Fig. 32. Object Face detection defects

Defect Description	Severity
The application is not able to detect numbers separated by commas.	Major
Appends unnecessary spaces at the end of the text result	Minor

Fig. 33. Text detection Defects

Defect Description	Severity
The application is not consistent in color detection.	Critical

Fig. 34. Color Detection Defects

Defect Description	Severity
The application is not consistent in color detection.	Critical

Fig. 35. Barcode Scanning Defects

6 Conclusion

Using Appium tool, we have written automation scripts for each individual AI feature of Envision AI application.Input to Envision app classification is necessary to identify all possible functional scenarios for testing of AI features:

- Object: Envision app camera can be directed at single or multiple objects when testing for objects in a scene.
- Human: Envision app camera can be directed at a human of either male or female gender for testing. The person in focus can be involved in any activity like reading,playing, walking.
- Text: Text in focus of Envision app can be either handwritten text or printed text. Text font size can be too large or too small or optimal. Text language under test is Hindi,English and numerical data. Text content can be a letter or a word or an entire paragraph.
- Color: The object in focus of Envision app can have a primary color or secondary color or tertiary color.
- Barcode: The barcode to be scanned by the Envision app can be present on a physical object or display screen of some device like a computer or tablet.