AIes (A project that can be used to helps the visually impaired people to interact with the world)

A Project Report
Presented to
The Faculty of the Computer Engineering Department

San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Bachelor of Science in Computer Engineering

By

Akash Sindhu
Maninder Singh
Jiawei Zhang
Tomas Chavez

04/2020

APPROVED FOR THE COLLEGE OF ENGINEERING

____
Dr. Gokay Saldamli, Project Advisor

____
Professor Rod Fatoohi, Instructor

____
Dr. Xiao Su, Computer Engineering Department Chair

ABSTRACT

AIes

By Tomas Chavez, Akash Sindhu, Maninder Singh & Jiawei Zhang

AIes is a web application model that will improve the daily lives of those that are visually impaired. This model is also integrated with the iOS application for users to use them. According to the World Health Organization, roughly 1.3 billion people in the world are afflicted by some form of vision impairment. AIes is able to classify and detect the objects in the images and videos that can be helpful after deploying on mobiles or devices which blind users will have access to.

The application is using a machine learning model to detect the objects in the images. Using a Convolutional Neural Network, a deep learning algorithm, the application identifies what objects are present in the images and videos. For the future use, we are going to implement a translational feature in the application that will convert the text to speech so that visually impaired users can understand. AIes model is not limited to work with iOS but can be extended to use in multiple platforms. AIes model will be available to all the developers so it can be integrated in any micro services.

On a broad level, there are three fields where we worked on this project. The first one is training a model using a convolutional neural network and achieving up to an acceptable accuracy. Having good accuracy is one of the biggest challenges. The second field is deploying the application using the flask so that it can be used for inference on new data. There were many small areas like learning CNN and hyperparameter tuning which we learned. As data is growing exponentially every day, we can understand the

relationship between data points and can develop better products using deep learning algorithms to help society. This app will help visually impaired people to better interact with the outside world just like a normal human being. The final field is developing an iOS Application, which allows users to take a picture or record a video on their cellular devices. In the backend, this will perform a POST request to our micro-service. The flask web server then will generate a json response and an image response that will be shown to the user.

Acknowledgments

[Your Acknowledgement statements are presented here].

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Project Goals and Objectives

Goals

- Implement and benchmark the state-of-the-art algorithms for object detection and localization and model deployment.
- Explore novel methods to deliver better models.
- Model deployment with flask web server and Heroku for inference.

Objectives

- Implement state-of-the-art convolutional neural network (CNN) for image.
- Narrate the objects and convert text to speech (if time allows).
- Reach the accuracy of at least 90% for image.
- Able to generate inference in real-time on new image and video.

1.2 Problem and Motivation

This project is important to visually impaired people and students who want to learn machine learning and computer vision state-of-the-art-algorithms. The motivation for this project comes from deep inside the heart to help the visually impaired people in this world and to learn new technological skills at the same time to prepare ourselves for the interviews and jobs. The needs of the project are having a good understanding of the machine learning algorithms, open source libraries, web servers, app development and integrating API (Application Program Interface) calls. Besides this, there is no hardware requirement for this project. Some of the problems we may encounter are related to the goals and objects we discussed above. Problems such as learning and implementing new libraries is a challenging task and can lead to many errors while integrating the web app with the saved model. Optimization to increase accuracy is a big challenge while developing machine learning models to achieve state-of-the-art results.

1.3 Project Application and Impact

The web application that we are developing can be useful in many ways. However, our primary goal is to deliver a http server url which will be used to develop iOS applications for visually impaired people. Besides this, our Flask server url can be integrated with other technologies for many different purposes where object detection can be useful in automation.

Specifically, we want our model to be used for visually impaired people as our dataset has 80 objects and our model can identify and label objects in an efficient manner. This will ensure that visually impaired people will be able to interact with the outside world in almost the same way as the normal person does. The application will allow visually impaired to have an idea of nearby objects using object detection algorithms. Students like us can also use this app to understand the underlying process, algorithms, and integration of different technologies to deliver the final product. This will motivate students to learn various new technologies that are used in the industry to make a difference in our society.

1.4 Project Results and Deliverables

The expected results of our project are:

- Able to achieve state of the art accuracy on new data.
- Able to run without any bugs.
- Able to detect objects in real-time (if time allows, considered as an extra feature).
- Code will be open-sourced on GitHub and it will be made available with the well explained documentation that will explain how to install and use it.

1.5 Project Report Structure

The project report will have 7 key chapters not including this introduction. It will include background and related work, project requirements, system design, system implementation, tools and standards, testing and experimentation, and conclusion with future work. Each section will address various principles of the projects. The background and related work will have the background and used technologies, literature survey and the state-of-the-art summary. The project requirements will include domain requirements, system (or component) functional requirements, non-functional requirements, context and interface requirements and technology and resource requirements. System Design will include architecture design, interface and component design, structure and logic design, design constraints, problems, trade-offs, and solutions. System implementations will have implementation overview, implementation of developed solutions, implementation problems, challenges, and lessons learned. Tools and standards have information pertaining to tools used and standards to be kept throughout the project. Testing and experiment include the testing and experiment scope, testing and experiment, approach, testing and experiment results and analysis. Finally, the conclusion with future work will close up our project and explain how the project can be continued.

Chapter 2. Background and Related Work

2.1 Background and Used Technologies

Object recognition is a general term for computer vision tasks that involve identifying objects in digital photos and videos. There are three phases in the object recognition process. The first one is image classification which predicts the class of object in the target image. The second phase is localization which identifies the location of multiple objects in the target image. Finally, the third phase is to show the bounding box with the highest Intersection Over Union (IOU).

The Convolutional Neural Network (CNN) is the algorithm in the field of computer vision in deep learning. A Convolutional Neural Network (CNN) is made up of different convolutional, pooling, and fully connected layers which result in high accuracy. This algorithm is useful because it learns the spatial relationship between the pixels of the image. Since, the images are made up of pixels and can have RGB channels, we have to find and learn the relationship between the pixels of the image. The convolutional layers are better for learning this because they share parameters many times and sparsity of connections which means each layer output depends on the small number of inputs (channel size), instead of taking into account all the inputs (pixels in the image).

Our program is using Flask web server, which is a web production server and framework that let developers make python web applications and deploy them. AIes will use Python as the primary language to create the machine learning model and deployment.

2.2 Literature Search

AIes will utilize the most recent research in machine learning. AIes will use a combination of object classification to classify which class the object belongs to, detection to detect the object location, localization to localize and make the bounding box around it. Intersection Over Union (IOU) and non-maximum suppression (NMS) will be used to get the final results. The convolutional neural networks are an active area of research because of these 7 reasons which makes CNN a best choice to do more research on are spatial exploitation, depth, multi-path, width, feature-map exploitation, channel boosting, and attention.

2.3 State-of-the-art Summary

Machine Learning has been a topic with a plethora of new research being discovered every day. As machine learning improves pattern recognition is being applied to more fields where human intervention was once needed. The health field has been early to adopt machine learning because of its ability to change peoples lives. There have begun attempts to improve the lives of those that are visually impaired. Some of the first attempts were made to help those that had difficulties reading because of their visual impairment. Research has been conducted that has made it possible to detect text in an image using "edge-preserving maximally stable extremal regions (eMSER) algorithm using the pyramid histogram of oriented gradients (PHOG)" (Joan, and Valli). Machine learning that can detect text could be very useful for multiple processes. It could be used for those who are visually impaired or even for many automation products.

There are beginning to be many applications that use similar algorithms to assist the visually impaired by identifying objects instead of text, but their effectiveness varies in different situations. This is the case when it comes to identifying objects. The objects identifiable are limited to the ones of which we have data. This creates the ability to identify objects that are very common, but difficult to identify more unique objects or newer objects. This is the case with such apps as "NantMobile Money Reader, LookTel Money Reader, and Eye Notes" (Weiss, Martin, Margaux Luck, Roger Girgis, Chris Pal, and Joseph Cohen.) where they can identify many different types of currency but cannot identify newly distributed bills. Newer applications have allowed users to scan personal objects to help the device identify them, but this is not very practical. Another process to note is that many of these applications rely on uploading the images to a server where the analysis is done, but this creates a latency.

There is much room for improvement with these applications. Recently improvements have been made in the identification of multiple objects in complex surroundings. Instance segmentation has become more efficient where objects are identified in a pixel region. Improvements have been made to classifying objects as well. Deep Convolutional Neural Networks implementation has outperformed traditional detection algorithms[8].

Chapter 3 Project Requirements

3.1 Domain and Business Requirements

UML Diagram:

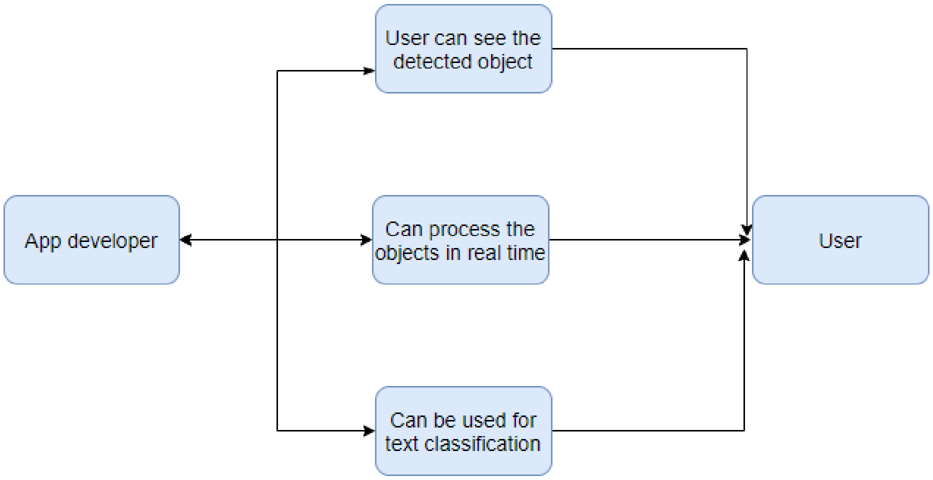

User can see the detected object
App developer
Can process the objects in real time
User
Can be used for text classification

Figure 1. AIes UML Diagram Flow

Sequence Diagram:

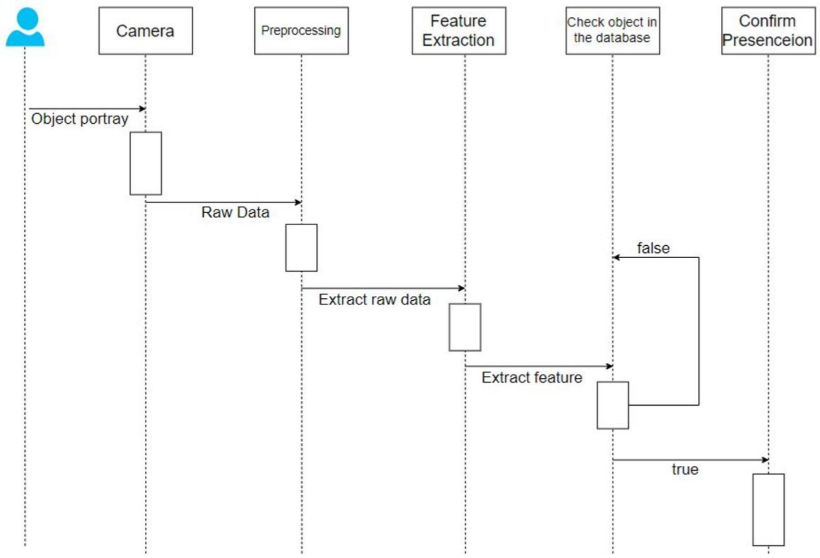


Figure 2. AIes Sequence Diagram Flow

Data Flow diagram:

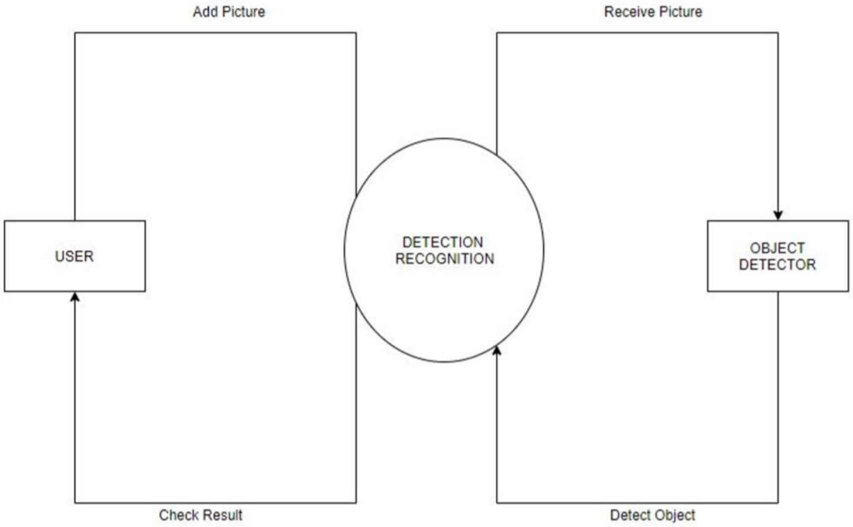


Figure 3. AIes Data Flow Diagram

3.2 System (or Component) Functional Requirements

The basic function of AIes project is to make an API endpoint which can be used to connect two software and help them communicate in real time. API endpoint will provide a url which does some functions like POST and GET based on the application's requirement. This url is used for communication between the backend where applications perform calculations with the frontend where the user interacts with the application to perform certain activities. We are using Flask web server with two app routes. One returns the classes predicted in JSON response and the other returns the images with detections. This url will be useful in the future, we may continue our work to implement the iOS or android app which will perform object detection with text to speech conversion.

Users

- Visually impaired people
- Students who are interested in machine learning

Use Case: Take images and videos to classify, detect and localize them in real time.

Summary: A user opens AIes project to get the API endpoint and integrate it with their use cases to generate real time object detection.

Actors: Application, Users

Description:

- Users download the github repository. Read the Readme file and perform required command line tasks.

- After the application is running, deploy the application to the cloud web microservice provider to get the permanent global usable API endpoint url.
- Open the Postmates and perform required POST and GET requests to test the API endpoint is working correctly.
- Use this url with their use case.

Essential Requirement

The requirements of the AIes project are very simple. Besides having the understanding of how machine learning algorithms work, how microservice architectures work, and python programming language, a user does not need anything else specifically to run this project according to their needs. If a user wants to access the model files and change them will require proper knowledge of deep learning and their libraries.

Desired Requirement

After the essential requirements, a user can have some desired requirements which will help him get things work more easily such as having experience with machine learning and deep learning concepts, their implementations, understanding of git, and general software engineering principles.

Future features

There are two future features that will include deciphering the JSON response from text to speech and recommending directions on the iOS application. The text to speech will use the Apple library called AVFoundation Framework. The direction suggestions will use both outdoor localization and indoor localization.

3.3 Non-functional Requirements

These requirements will not specify the behavior of our project and only be used to judge the operation. Some of the non-functional requirements are scalability, maintainability, serviceability, and security. The user needs to be connected to the internet, which is necessary as of now.

3.4 Context and Interface Requirements

Supportability:

- AIes Flask web server is developed in Python
- AIes iOS Application is developed in Swift

Interface:

- The system shall be accessible on all iOS devices running iOS 12.0 and above using camera.

Packaging:

- AIes project will be open-sourced on GitHub.

Frameworks and Languages used in the development:

- Python: A top-level language, general-purpose programming language.
- Tensorflow: A machine learning library.
- Keras: A high level machine learning library.
- Flask: A micro web framework written in python.
- CocoaPods: is a dependency manager for iOS and Swift projects. It contains over 65 thousand libraries that can help towards our project.

- Apple UIKit: Used to provide the required infrastructure such as window and view architecture for implementing interfaces.
- Apple AVFoundation: Used in areas regarding photo and video captures from iOS devices that will be required to send the data to cloud.
- Apple MobileCoreServices: Required to allow manipulating local data present on phones such as photos and videos to process.
- Heroku: Platform that allows the ability to deploy Flask web server.

3.5 Technology and Resource Requirements

	Front End	**Back End**
Language	**Swift**	**Python**
Framework	**AVFoundation, UIKit, Cocoa Touch, MobileCoreServices**	**TensorFlow, Keras, numpy, opencv-python==4.1.1.26, lxml, tqdm, flask, seaborn, pillow, gunicorn, opencv-contrib-python,**
Version Control System	**Github**	**Github**
Software Compiler	**Xcode**	**Pycharm**

Table 1: Technology requirements

Chapter 4 System Design

4.1 Architecture Design

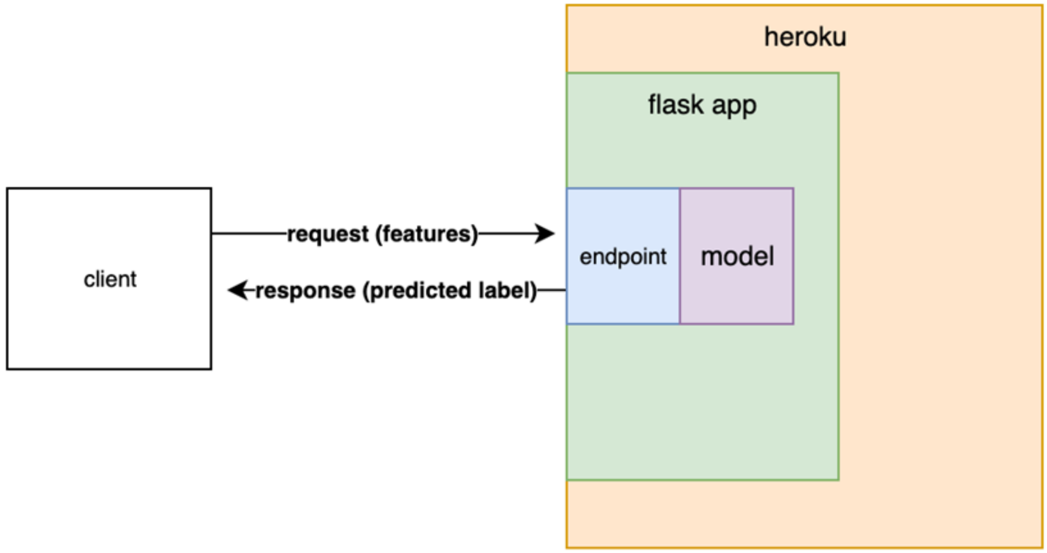


Figure 4: Client Server Connection with Machine Learning Model

4.2 Interface and Component Design

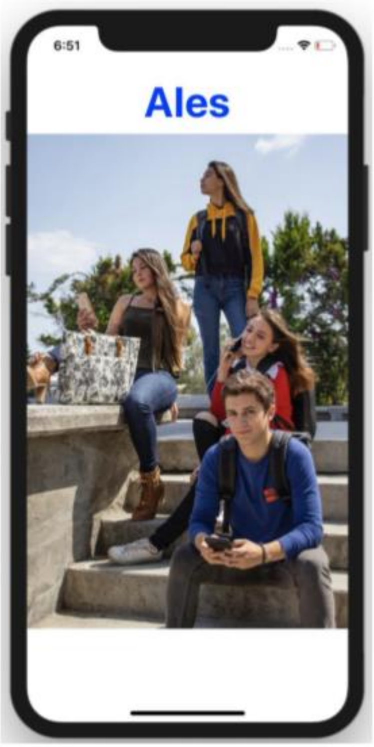


Figure 5. AIes User Interface Design

4.3 Structure and Logic Design

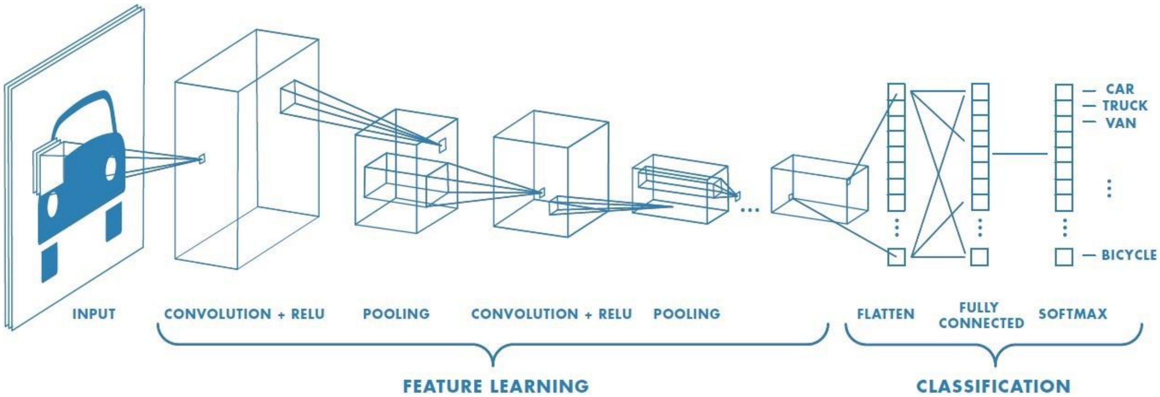


Figure 6. Convolutional Neural Network Demonstration

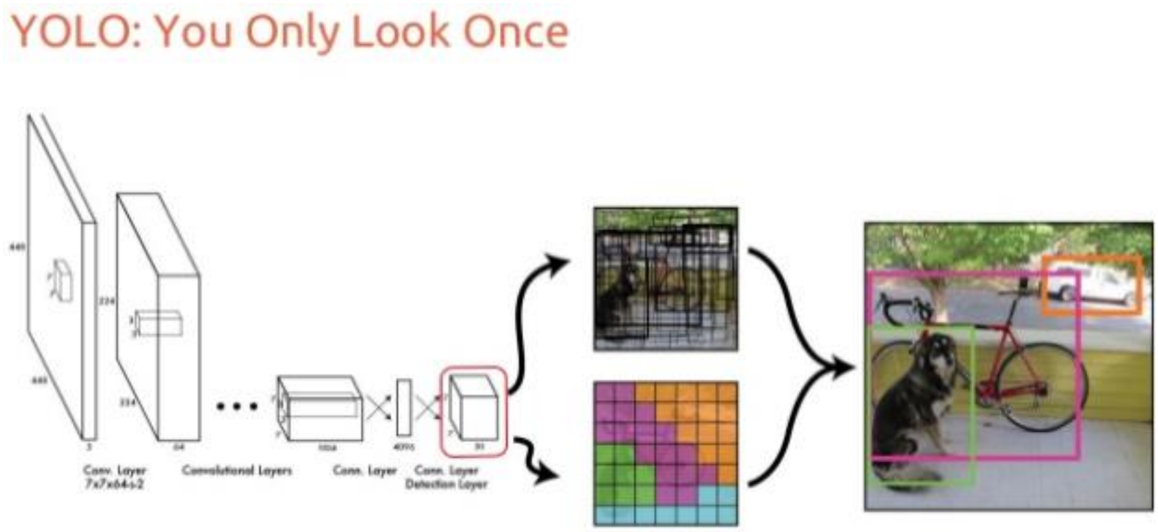


Figure 7. You Only Look Once: Real Time Object Detection

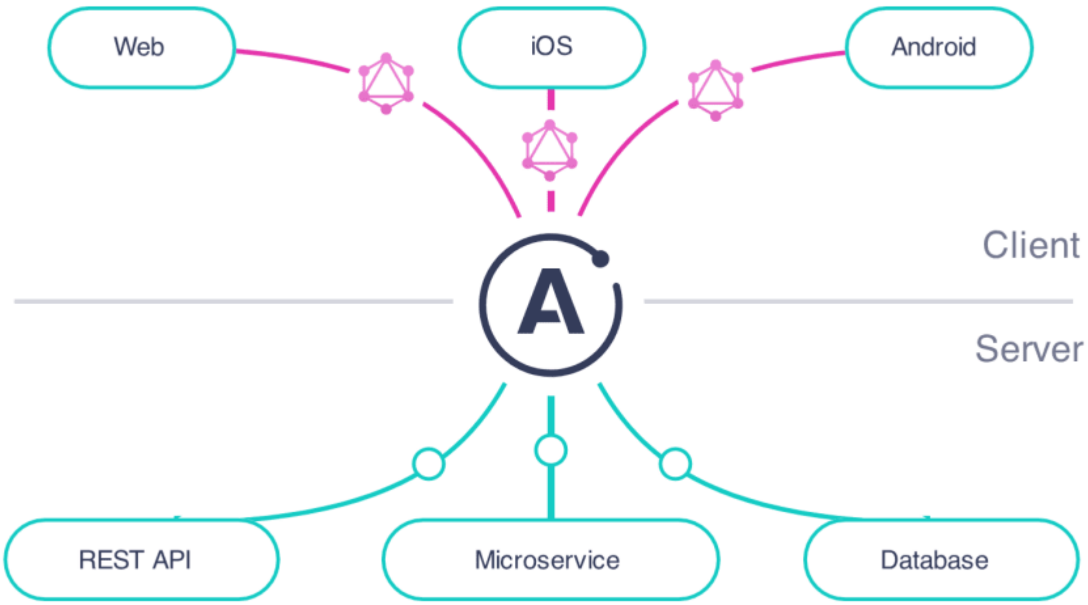


Figure 9. Client Server Connection

4.4 Design Constraints, Problems, Trade-offs, and Solutions

4.4.1 Design Constraints and Challenges

In this section, we compare the approaches and their constraints we discovered during this project. We will discuss the advantages and disadvantages of local client side models versus cloud machine learning models.

Advantages of Local Machine Learning mode:

Speed - Accessing data stored on client devices is much quicker then calling the API on cloud services.

Security Control - The system has full control over the security of the data. Especially the images taken from the user should be encrypted.

Disadvantages of Local Machine Learning model:

Flexibility - Local Machine Learning model. Having a local stored machine learning model means that we have to worry about upgrades and maintenance for each type of device, which can be complicated in later development.

Advantages of Cloud Machine Learning model:

Data Retrieval - Data can be accessed from anywhere as long as the user's device has a network.

Flexibility - Cloud computing model does not have to worry about upgrades and maintenance.

Disadvantages of Cloud Machine Learning model:

Security and Privacy - It is challenging to keep data integrity when transmitting the data back and forth through the host and cloud platform.

Some of the other design constraints are time, subject matter expert availability, technology limitations and learner motivation. Design constraints include non-functional requirements which are constraints placed on the system or on the development process.

4.4.2 Design Solutions and Trade-offs

The design constraints have solutions as well and because we want our project to be more user-friendly, we spent a lot of time in understanding the difference between the end-user needs and developer's needs. That's why we decided to deliver our project in the form of an API endpoint which can be used for any kind of object detection use case.

Besides this for our iOS app, there are few UX design constraints that every mobile product needs to overcome. We will make sure that our design is simple and user friendly. We will label buttons and frames, so their purpose is clear. The phone will have the added difficulty of having a user interface that can be interacted by someone with limited vision. Our app will overcome this by using very few buttons that can easily be located when holding the mobile device. The buttons will be located near the bottom of the device on both the right and left side. A sound will also be made as an image is taken to alert the user.

The biggest and most important constraint is Client-Side storage, meaning users having less space on their mobile devices. As most of our development code will be working in Heroku server. The client side will have the implementation of using a camera as the hardware, but computations will be handled using Flask server. Whenever a machine learning model is sent into production, the most important constraint is how to get precise results. In our case, we want to get accuracy of more than 95%, for that we need to do a lot of hyperparameter tuning and optimization using different techniques like early stopping and right kind of regularizers.

Our app will only work within the iOS because we are not making it available for android. This will limit the use to people who have only Apple phones. We might launch an app on google store in future if we get user reviews to make it available for Android. For our app we are not considering any special hardware requirements except camera and an internet connection. A proper documentation will be provided as soon as possible after launch and feedback will be welcomed so that we can upgrade the app and documentation. For error handling and extreme conditions, we are not going to consider our first version, later versions will be more compatible for a large number of user requests.

Chapter 5 System Implementation (Akash Sindhu)

5.1 Implementation Overview

In order to make this project viable and work properly we have divided the project into 4 parts: data collection and analysis, model training, model deployment and iOS application. For the data collection we took the MSCOCO dataset which has 80 different objects and their annotations. For the model training, we used a CNN (Convolutional Neural Network) algorithm which is basically an object detection algorithm with multiple layers of neural network, in our case we used ResNet-50 (50 layers) model structure. We then deployed this on the Heroku server to get the endpoint so that we can use API Gateway and Flask for inference in real time from the iOS application. For the iOS application we used the basic features like camera having capabilities to take pictures and video.

5.2 Implementation of Developed Solutions

We choose to use the Object detection algorithm with ResNet-50 architecture because it performs the best and has deep layers stacked up which can learn things much better than anything other structure we came across. Deep networks are hard to train because of the notorious vanishing gradient problem — as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient infinitely small. As a result, as the network goes deeper, its performance gets saturated or even starts degrading rapidly.

The ResNet has a stack of convolutional layers, pooling layers and fully connected layers. Resnet is just a combination of all of them in such a way that we are able to skip every other layer so that we can deal with vanishing gradients without any loss.

After the model was trained locally, we deployed it with flask webserver and got the endpoints. We used microservice architecture so that it can be scaled, and we don’t have to worry about the infrastructure and cost of it.

5.3 Implementation Problems, Challenges, and Lesson Learned

Over the course of the project we have encountered many challenges including two major ones. The two challenges were making the data ready for the algorithm and learning Flask integration.

Making the data ready for the algorithm:

We were having difficulty in accessing and preprocessing the data in the specific format we are required for the algorithm to understand it. We were able to successfully annotate the images and pass them to the model.

Learning Flask and Integration:

None of our group members have had experience using machine learning model deployment using flask. Testing out our live API application was also a bit challenging because we were not experienced with such kind of model integrations.

Chapter 6 Tools and Standards

6.1. Tools Used

There are several tools we have used for this project which helped us in completing the project and develop a streamlined flow of tasks within the team and project. Some of the tools we are going to list were discovered, learned and used to complete this project on time. Without these tools, we would have not completed or partially completed the project.

Since our project is all about machine learning, we heavily relied on many tools to contact the teammates, discover new ways to solve errors and implement and deploy our product to the end-user.

Flask

For deploying our application, we used Flask in order to provide an endpoint API gateway for future applications. Flask is a web server and framework that integrates the model files with app route files to serve them on a server on a specific port. We then can deploy this on cloud to achieve real time object detection with low latency.

GitHub

We have used the Git and GitHub service to organize our code and keep track of where we are so that all the people in the team are in sync.

iOS Development

For iOS development, we have used GitHub. GitHub was chosen because it is the version control that our group is most familiar with. It works very well in resolving conflicts. Even though GitHub is a great tool it is difficult to use with storyboards. Because of this we have limited our use of storyboards and have mostly coded the user interface.

Xcode is the application that our group is using for iOS application development. Not only does it allow us to use simulators to cross check the various Apple devices, but also allows us to design the user interface using a storyboard, running and debugging the code. Xcode will also be used in future when we are distributing the application onto the iOS App Store. Xcode's Interface Builder is the editor we are using to assemble our app's user interface with the help of preconfigured objects that are provided in the Library. The objects include windows, text fields, buttons and views.

Frameworks

We have used a lot of different kinds of libraries and frameworks, but mainly we used TensorFlow and keras to implement and train our model. For the deployment we used flask, postman to test our model. This was the best learning curve for us, as we didn't have exposure to a lot of libraries which we used in our project. This was new to us and we ended up using them properly.

6.2. Standards

From the machine learning perspective, there are not a lot of standards we need to follow to deliver something as it is the constant iterative process to update the data, model, and features whether they are about the data or how scalable the system is.

Data

The data we used for our model is the COCO dataset provided by Microsoft. It is the dataset with 80 objects and provides annotations of each of the images so that it can be used for object classification, detection, and segmentation. We used this dataset because it is open-sourced and easy to use and provides state-of-the-art results.

Accuracy

Accuracy is one of the best things we can discuss machine learning problems since the output and results are highly dependable on the accuracy of the model which defines if the model will be made available for the public to use. For our model, we have decided to put an extra effort into this so that the end-user can use and trust our project. For our model, all the labels (objects predicted) have an accuracy of more than 95%.

Frameworks

The documentation of all the frameworks is followed. The guidelines indicated each specific step for a developer to create and manage machine learning models.

iOS Development

Apple's user interface guidelines have been followed. These guidelines exist to make sure Apple applications have similar user interfaces and allow for optimal user experience. The code structure is also structured in the Model-View-Controller structure.

This allows for the code to be well organized and only have the correct controller have pertinent information.

Chapter 7 Testing and Experiment

7.1 Testing and Experiment Scope

We had two broad areas for testing and experiment scope. Below are the specific tests we carried out for our project. Machine learning models are gone through various testing procedures to see if they are able to calculate the model accurately or not. In our case, since we used the pre-trained weights of the yolov3 model, we are not bound by this requirement much. Still there are some tests we considered to include which shows the total time required to generate prediction by the model, accuracy of the model on images and videos.

For the iOS application we checked on many things such as application crashing, application incompatibilities, security vulnerability, memory leaks, etc. Many applications crash due to some internal bugs or memory leak in apps. Memory leaks cause applications to crash. In our case, we made sure that we are using a limited amount of memory of client-side storage and memory because we are using the client's camera as hardware, and operating system as our software.

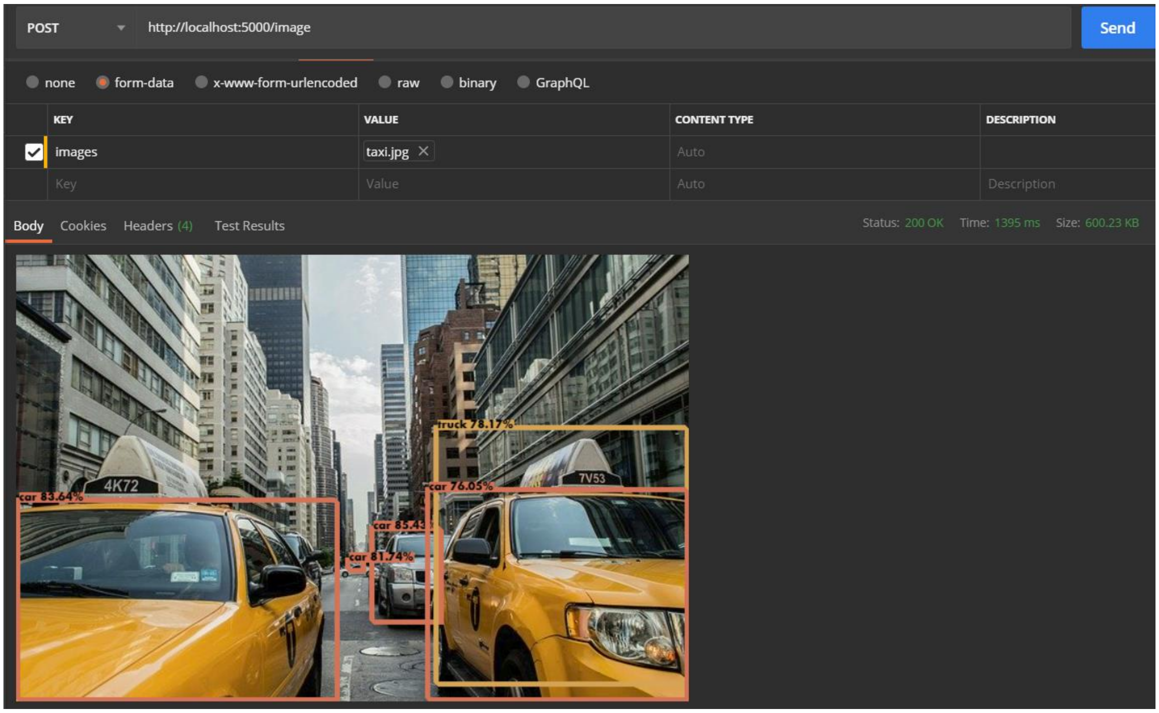
Some of the testing checklists we followed made sure that our application had an app icon and name. Also, make sure that both icon and name are self-explanatory reflecting the core intention of the application. Checked the splash screen timeout and made sure to check our application asking for user permissions to access the camera and microphone for our features. Our application only supports portrait mode, so no orientation checks were checked.

7.2 Testing and Experiment Approach

Since, there are limitations on testing in machine learning, as it is not the same as software testing. In machine learning models, we first have to train the model and then evaluate on development data. This gives us the accuracy of the model which shows if the model is overfitted, underfitted or works perfectly. There are many ways to check the accuracy like confusion matrix, RMSE, F-1 score, recall, and precision. But in our case, we didn’t evaluate the model as there is no need for it because we used pre-trained model weights. The weights were already trained on so 80 different images. We tested our loaded model on new data and checked if it is working or not as it should. We found out that our model is performing well.

For iOS user interface testing, Unit Testing was used. Unit Testing was split up based on the different buttons to be tested. The user interface was also examined by pressing buttons while simulating being visually impaired to determine if the testing would be adequate for our clients.

7.3 Testing and Experiment Results and Analysis



POST http://localhost:5000/image Send

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	CONTENT TYPE	DESCRIPTION
☑	images	taxi.jpg ✕	Auto	
	Key	Value	Auto	Description

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 1395 ms Size: 600.23 KB

Figure 10: Image detections with total time taken

Figure 10 clearly shows the post request sent to the web server through the route /image. Every object in the image is identified with good accuracy which shows that our project is working and can be used for future use. In the image on the right side, we can see that Time taken by the Postman to send the request and get the response is around 1395 ms which is very good for real time solutions.

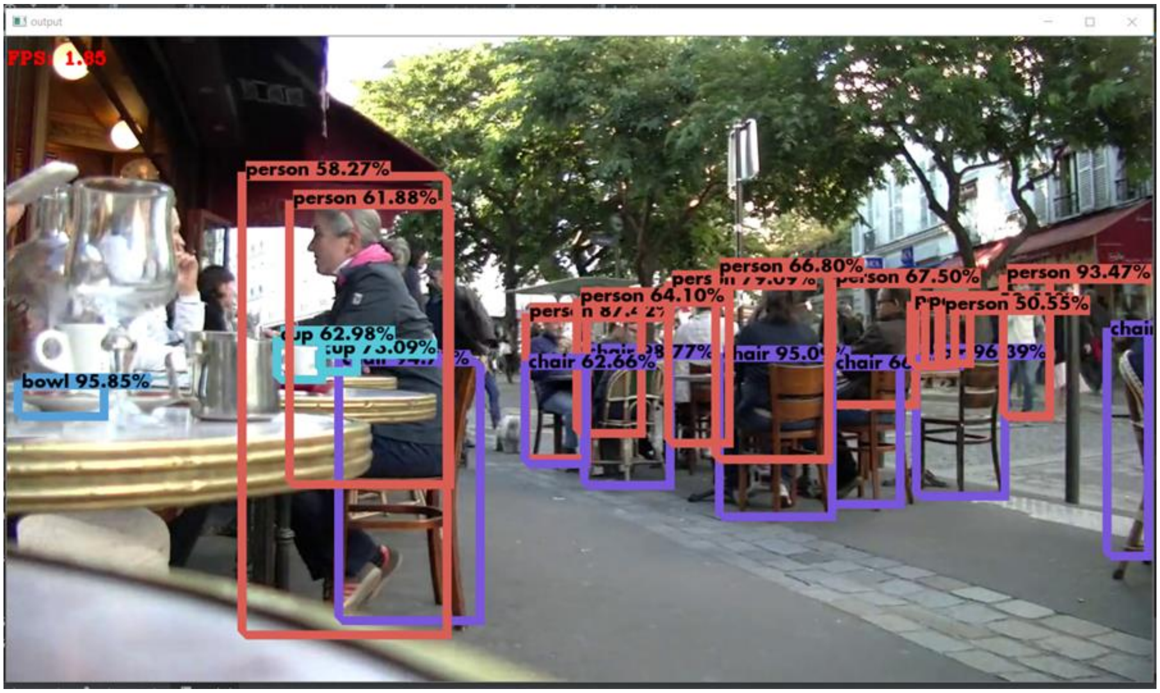


Figure 11: Video showing the FPS and objects detected in real time.

Besides this, we also test our project on videos. Our tests are good in video and we are able to generate inferences in real time. From Figure 11, we can say that our model is ready to be used for video purposes also. We have not done any specific testing like A/B testing so far. We have tested our model and it works well with all the 80 objects it was trained on. The API endpoint we created is doing very well on real-time predictions and this made our project very user-friendly for those who don't have specific knowledge about machine learning and want to make applications.

iOS User Interface Testing:

Photo Button	Video Button	Text To Speech Button
Correctly saved image	Correctly saved video without audio	Repeated Text Correctly

Note: Photo button and video button were tested in various orders with correct results

Chapter 8 Conclusion and Future Work

In conclusion, we will discuss what we have achieved, learned and will do in future after completing the project. Besides this, we will talk about what are the possible solutions, improvements, and use cases for our project in the future. As our motive was to deliver the project in such a way that the end-user can re-use it without knowingly a lot of machine learning. We are able to achieve this by generating API endpoint url that can take images, and videos to generate JSON response of objects identified and images with objects localized on it. This is a great achievement for our project and we are proudly developing this project on GitHub to open-source it.

Besides this, our project is mainly developed for the use case of making an application for visually impared people so that they can better interact with the world around them. We are able to achieve this as we have delivered the API gateway url, which people can directly use for making an app for visually imapred people. We have made it so easy for everyone to connect our API with their application and run it within 15 minutes. This is even bigger achievement for all of us because we never worked on anything like this before where we had to be in contact with the end-user who wants to use our service considering the difficulties developers had to go through that includes the time to develop new applications and their re-usability.

In the end, we would like to say that object detection is an area of improvement which focuses on many aspects to achieve state-of-the-art accuracy. For our future improvements, we would like to work on one of the application use-cases where we will deliver it to the end-user.

Future work

Despite the achievements, our project can still be improved and below is a list of direction:

1. Since we are able to generate API endpoints for our project, we are going to deliver an actual iOS or Android app for visually imapred people.
2. AIes machine learning algorithm can detect up to 80 different classes of objects. In the future, more classes can be targeted by training with a larger dataset.

References

1. Caixinha, Miguel, and Sandrina Nunes. "Machine Learning Techniques in Clinical Vision Sciences." Current Eye Research 42.1 (2017): 1-15. Web.
2. Catalunya, U. P. de. . Object Detection (D2L5 Insight@DCU Machine Learning Workshop 2017).(2017) Web.
3. Chris . Productionize a Machine Learning model with Flask and Heroku. (2019) Web.
4. Joan, and Valli. "An Enhanced Text Detection Technique for the Visually Impaired to Read Text." Information Systems Frontiers 19.5 (2017): 1039-056. Web.
5. Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*. doi: 10.1007/s10462-020-09825-6
6. Saha, S. . A Comprehensive Guide to Convolutional Neural Networks-the ELI5 way. (2018) Web.
7. "Patent Issued for Utilizing A Machine Learning Model To Automatically Visually Validate A User Interface For Multiple Platforms." Journal of Engineering (2018): 624. Web.
8. Weiss, Martin, Margaux Luck, Roger Girgis, Chris Pal, and Joseph Cohen. "A Survey of Mobile Computing for the Visually Impaired." ArXiv.org (2018): ArXiv.org, Nov 27, 2018. Web.
9. Xiongwei Wu, Doyen Sahoo, Steven C.H. Hoi. "Recent Advances in Deep Learning for Object Detection." Neurocomputing (2019). Web.