**PROBLEM STATEMENT**

The learned camera is an android application which primarily focuses on helping the visually disabled people by giving them information about their surrounding objects by either classifying or identifying the objects.

The app can simply be used by pointing out the device’s camera towards the object and the application will respond by either classifying or identifying the object. In addition to this, the android application will also speak out the obtained result. The algorithms written to predict the given image comes under Machine Learning (ML). The model obtained after applying these algorithms will be stored in a file which will allow us to use this model to obtain the predicted result. A server-client connection will be established. The client will send an image to the server which will be processed and the result will be returned back to the client.

**Technologies used:**

* Python
* Java
* Flask
* Volley
* openCV
* Keras
* Matplotlib.pyplot
* Numpy

**Software:**

* Anaconda Distribution
* Android Studio
* Ngrok
* PyCharm

**Team Members Details And Their Contribution:**

|  |  |  |  |
| --- | --- | --- | --- |
| S no | Enrollment Number | Student Name | Contribution |
| 1 | 15103298 | Khyati Sehgal |  |
| 2 | 15103302 | Mohd. Shuja Reshi |  |
| 3 | 15103304 | Prafull Pandey |  |
| 4 | 15103310 | Tanya Gupta |  |

**Novelty of added functionalities:**

Face detection and Object detection could of great use to those who are visually impaired. It can be of great help to them if such an application is present in their mobile devices , with a simple user interface , which is with them everywhere and could assist those people by interacting with them i.e. speaking out the detections.

**Initial State of the Project :**

This android project is a camera based application. It captures picture using the device camera. Captured picture gets saved in device’s internal storage. By using implicit intent, it has the ability to access different picture clicking applications. It also provides a simple feature of recapturing the clicked picture as per the user’s desire.

**Final State of the Project :**

The original android application was simply clicking the picture using the device camera triggered on the click of a button which is finally replaced with a camera view using flurgle camera toolkit available open sourced on github. We have further provided some smartness to the device camera using Machine Learning algorithms.

**Functionalities Added :**

* **OBJECT DETECTION**

This functionaltity allows the user to provide the system with an queried image and get it identified. The object classified in this functionality cover the classes of cifar dataset. We will be applying Convolutional neural network (CNN) algorithm , the

input image is convoluted ,pooled, flattened . This will be presented as an input layer to the hidden layers which are fully connected weighted graph. At the end of it, it has an output layer that finally classifies the given input image as per the cifar dataset.

* **FACE RECOGNITION**

This functionality will be used to recognize the face of a person. OpenCV will be used for capturing and detecting the faces of the user in order to extract the features from the face. Further we will be using the K-nearest neighbor (KNN) algorithm in which Euclidean distance between the input image and the obtained dataset is calculated. The input image is classified by a majority vote of its neighbors, with the image being assigned to the class which is most common among its k nearest neighbours ,ultimately recognising the face.

* **TEXT-TO-SPEECH**

The user can hear out loud the output of the above recognition – face or the object.For this we will be using the inbuilt text to speech library of android to convert text to speech. This could be a great help to visually impaired people.

**Algorithms Used :**

1. **CONVOLUTIONAL NEURAL NETWORK (CNN)**

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), a convolutional neural network (CNN, or ConvNet) is a class of deep, [feed-forward](https://en.wikipedia.org/wiki/Feedforward_neural_network) [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network)that has successfully been applied to analyzing visual imagery.

CNNs use a variation of [multilayer perceptrons](https://en.wikipedia.org/wiki/Multilayer_perceptron) designed to require minimal [preprocessing](https://en.wikipedia.org/wiki/Data_pre-processing). They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and [translation invariance](https://en.wikipedia.org/wiki/Translation_invariance) characteristics.

CNNs use relatively little pre-processing compared to other [image classification algorithms](https://en.wikipedia.org/w/index.php?title=Image_classification_algorithm&action=edit&redlink=1). This means that the network learns the [filters](https://en.wikipedia.org/w/index.php?title=Filter_(image_processing)&action=edit&redlink=1) that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in [image and video recognition](https://en.wikipedia.org/wiki/Computer_vision), [recommender systems](https://en.wikipedia.org/wiki/Recommender_system)and [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing).

A CNN consists of an input and an output layer, as well as multiple [hidden layers](https://en.wikipedia.org/w/index.php?title=Hidden_layer_(neural_network)&action=edit&redlink=1). The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers.

* Convolutional

Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli.

Each convolutional neuron processes data only for its receptive field. Tiling allows CNNs to tolerate [translation](https://en.wikipedia.org/wiki/Translation_(geometry)) of the input image (e.g. translation, rotation, perspective distortion)

Although [fully connected feedforward neural networks](https://en.wikipedia.org/wiki/Multilayer_perceptron) can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in a shallow (opposite of deep) architecturedue to the very large input sizes associated with images, where each pixel is a relevant data point. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. In other words, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using [backpropagation](https://en.wikipedia.org/wiki/Backpropagation).

* Pooling

Convolutional networks may include local or global pooling layers, which combine the outputs of neuron clusters at one layer into a single neuron in the next layer. For example, *max pooling* uses the maximum value from each of a cluster of neurons at the prior layer. Another example is *average pooling*, which uses the average value from each of a cluster of neurons at the prior layer

* Fully connected

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network ([MLP](https://en.wikipedia.org/wiki/Multilayer_perceptron)).

1. **K-nearest neighbor (KNN)**

The k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression :

* In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.
* In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

**FUTURE SCOPE :**

In future we will be looking forward to extend our application in following areas :

* **Dataset**
* Extending the present CIFAR-10 dataset which consists of only 10 classes to much wider dataset of CIFAR-100 or ImageNet dataset.
* **OpenCv**
* Rather than clicking the picture to detect an object , the app will be deployed using OpenCV which will help to continuously capture each and every frame and hence identify the object without any external user interference.
* **Augmented Reality**
* In addition to deep learning algorithms , Augmented Reality can also be added as a functionality which can determine the distance of object from the user.
* **Offline**
* Making the application completely offline by storing the model in android application itself and detecting objects from the app without going through client-server connection.

**REFERENCES**

[**http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/**](http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/)

[**https://www.learnopencv.com/image-recognition-and-object-detection-part1/**](https://www.learnopencv.com/image-recognition-and-object-detection-part1/)

[**https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/**](https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/)

[**http://colah.github.io/**](http://colah.github.io/)

**Additional Reading :**

Neural Networks and Deep Learning

By : Michael Nielsen (2015)

Introduction to Convolutional Neural Network

By : Jianxin Wu(2017)