Phone2Cash

Duvvuri Venkata Sai Sri Hari
Software Engineering
San Jose State University
San Jose, USA
duvvurisrihari@gmail.com

Farha Kauser
Software Engineering
San Jose State University
San Jose, USA
Farha.kauser1@gmail.com

Vamsi Chakravartula
Software Engineering
San Jose State University
San Jose, USA
vamsikrishnachakravartula@gmail.com

Vishal Gadapa
Software Engineering
San Jose State University
San Jose, USA
vishalgadapa2468@gmail.com

***Abstract*— A streamlined process of turning in our old mobiles in exchange for cash would significantly simplify the hassle of online search of prices and negotiations for a fair deal. A proposed solution to this could be a vending machine that takes the phone, scans it and reckons the price of it. If satisfied with the price, a cheque could be dispensed for the equivalent amount offered. As a part of the solution, an application is to be designed, which takes the images of a damaged phone as input and quantifies the value of the phone based on the damage to the screen. Instead of a fixed price system that provides static output prices, without the consideration of the state of the phone, an ingenious system that estimates based on the condition of the phone can be developed. To pull this off, a trained machine learning model is used to analyze the data set of captured images. The model is initially trained with data of various broken phone images. An interactive user interface with ReactJS, secured with authentication factors is designed for the ease of the customer where the model of the phone must be entered to gauge the original cost of the phone. Based on the results, the percentage of cracks, combined with the standard model prices, the final price of the mobile can be predicted.**

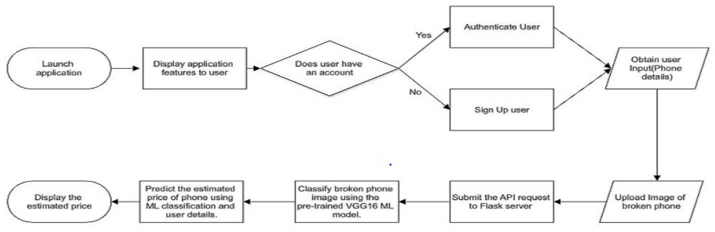
Keywords—ReactJS, Machine Learning

I. Introduction

Mobile Phone is the most important tool for survival in this world. Every function is related to the use of a phone What if the Phone breaks. Life becomes miserable. It is not easy to buy a new phone instantly due to economic reasons. We can't discard the old phone as well. The solution to all these problems is an application. This application estimates the cost of a phone by considering its screen condition, whether broken or intact and its original price. The present situation for a person whose phone is broken is to go from one shop to shop and enquire about the phone exchange. There is no guarantee that the deal that they obtain is fair. But this tool takes the input of the phone images and assess the quality of the phone state and estimates the price of the phone. So, this reduces the effort of the user and increase ease of access. This decreases the stress to the user. Also, the phones can be reused avoiding the wastage and damage that is caused by simply dumping. Thus, ensuring trash to cash

A. Basics of Machine learning

The Machine learning model used is VGG16 convolution neural networks. This model is mainly used for the classification of the images. The model is initially trained based on the inputs of broken phone screens. First, the dataset is divided into three categories. The first category is the training dataset. This data is used to train the model. The data present in this model is crucial as this decides the working of the model. The second dataset is the validation dataset which validates the model at various steps. If the validation check fails, the model retrains based on the data. The validation dataset can be used as a check for assessing the working of the model. The third dataset is the testing dataset. The testing dataset is the main deciding factor for the complete check of the model. Generally, the dataset is divided into seventy and thirty ratios for the training and testing datasets respectively.


Launch application
Display application features to user
Does user have an account
Yes
No
Authenticate User
Sign Up user
Obtain user Input(Phone details)
Upload Image of broken phone
Submit the API request to Flask server
Classify broken phone image using the pre-trained VGG16 ML model.
Predict the estimated price of phone using ML classification and user details.
Display the estimated price

B.Application working

An application is built using the ReactJS and NodeJS. The overall front end is written in ReactJS. The user is asked to Sign up or Log in into the application using his credentials. They can even login with their Gmail or Facebook logins as well. Upon login, the user is presented to enter the details of his phone. After that, he is asked to enter a few vital questions regarding the working of the phone.
Based on the inputs by the user, this image is uploaded to the server, which undergoes classification based on the pre-trained VGG Convolution model. The classified image is classified based on the number of cracks present on the screen and then is classified into one of the categories.

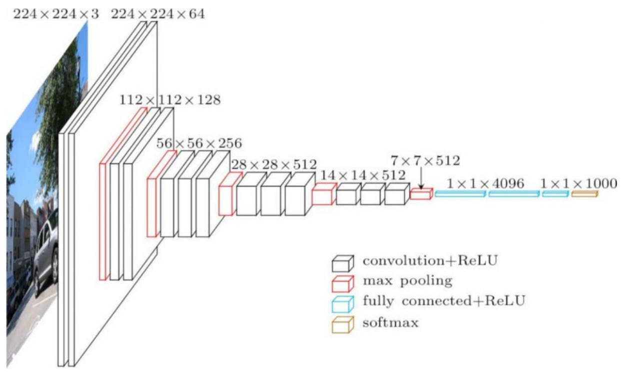
Considering all the inputs the price of the phone is estimated.

C. VGG 16 Convolution neural network model

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014.

It improves over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black Gunshot to use abbreviations in the title or heads unless they are unavoidable. The input to the cov1 layer is of a fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filter, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max pooling). Max-pooling is performed over a 2×2-pixel window, with stride 2.

Three Fully Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks. [1]

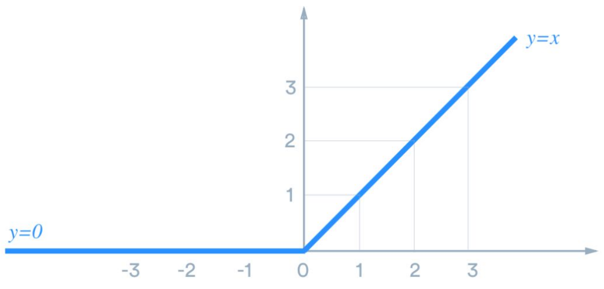


VGG16 Architecture

All hidden layers are equipped with the rectification (ReLU) non-linearity. It is also noted that none of the networks (except for one) contain Local Response Normalisation (LRN), such normalization does not improve the performance on the ILSVRC dataset but leads to increased memory consumption and computation time.

D.Rectified Linear Unit

ReLU stands for rectified linear unit and is a type of activation function. Mathematically, it is defined as y = max(0, x). Visually, it looks like the following:



ReLU is the most commonly used activation function in neural networks, especially in CNNs. If you are unsure what activation function to use in your network, ReLU is usually a good first choice.

ReLU is linear (identity) for all positive values, and zero for all negative values. This means that:

- It’s cheap to compute as there is no complicated math. The model can therefore take less time to train or run.
- It converges faster. Linearity means that the slope doesn’t plateau, or “saturate,” when x gets large. It doesn’t have the vanishing gradient problem suffered by other activation functions like sigmoid or tanh.
- It’s sparsely activated. Since ReLU is zero for all negative inputs, it’s likely for any given unit to not activate at all. This is often desirable[2]

F. Softmax

Softmax function, a wonderful activation function that turns numbers aka logits into probabilities that sum to one. Softmax function outputs a vector that represents the probability distributions of a list of potential outcomes. It’s also a core element used in deep learning classification tasks

In mathematics, the softmax function, also known as softargmax or normalized exponential function is a function that takes as input a vector of K real numbers and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. That is, prior to applying softmax, some vector components could be negative, or greater than one; and might not sum to 1; but after applying softmax, each component will be in the interval (0,1), and the components will add up to 1, so that they can be interpreted as probabilities. Furthermore, the larger input components will correspond to larger probabilities. Softmax is often used in neural networks, to map the non-normalized output of a network to a probability distribution over predicted output classes.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$$

In words: we apply the standard exponential function to each element of the input vector and normalize these values by dividing by the sum of all these exponentials; this normalization ensures that the sum of the components of the output vector is 1. Instead of e, a different base b > 0 can be used; choosing a larger value of b will create a probability distribution that is more concentrated around the positions of the largest input values[3][4][5]. Writing or [a] (for real β) [b] yields the expressions: [c]

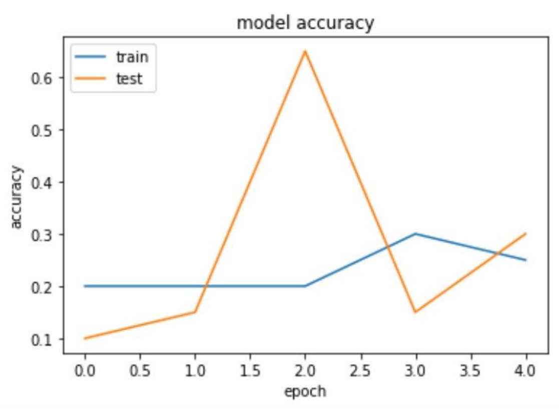
$$\sigma(\mathbf{z})_i = \frac{e^{\beta z_i}}{\sum_{j=1}^{K} e^{\beta z_j}} \text{ or } \sigma(\mathbf{z})_i = \frac{e^{-\beta z_i}}{\sum_{j=1}^{K} e^{-\beta z_j}} \text{ for } i = 1, \dots, K.$$

II. Working Model results

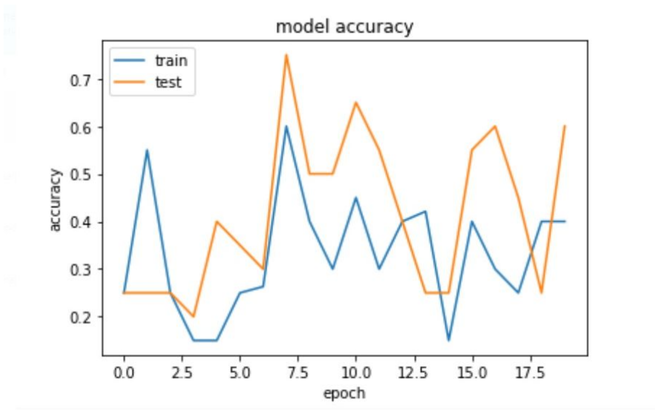
The model predicts the percentage of cracks on the screen. The graphs depicted below indicate the accuracy of the model with the variation in epochs. An increase in the number of epochs will increase the accuracy but further increase in the accuracy will over the model.

The test accuracy versus the number of epochs depicts the accuracy.

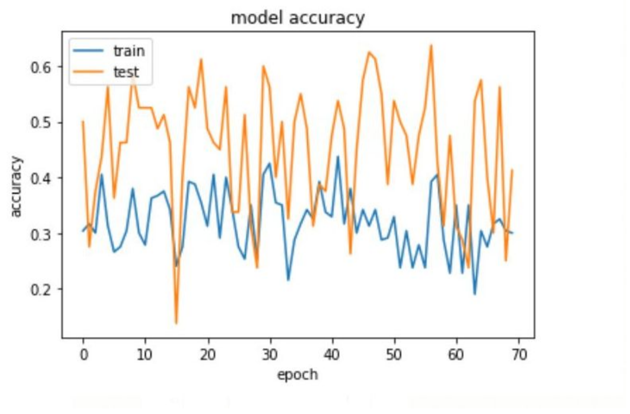
For the epochs = five, the plot is as follows



For the epochs = seventeen, the plot is as follows



For the epochs= seventy, the plot is as follows



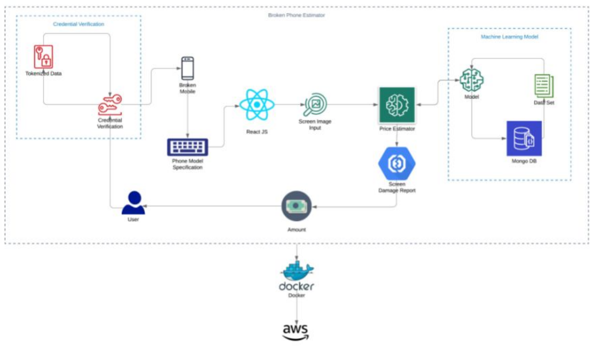
These results indicate the accuracy of the model for the data that was provided.

III. Application implementation

An application that estimates the cost of a phone by considering its screen condition, whether broken or intact and its original price.

A streamlined process of turning in our old mobiles in exchange for cash would significantly simplify the hassle of online search of prices and negotiations for a fair deal. A proposed solution to this could be a vending machine that takes the phone, scans it and reckons the price of it. If satisfied with the price, a cheque could be dispensed for the equivalent amount offered.

The component diagram of the application is as follows.



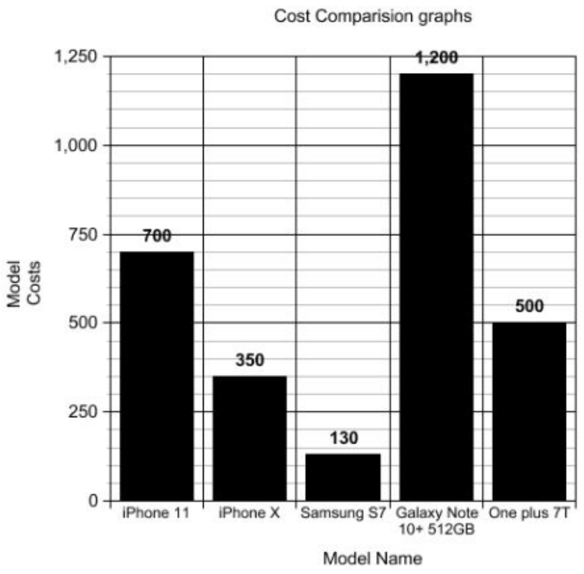
A.Technology Stack

Programming languages & Libraries: JavaScript, Python, Keras, NumPy, Panda
Technologies: Machine Learning, Computer Vision, Deep Learning
Web Technologies: HTML5, CSS3, JavaScript, React, Redux, Node.JS, REST API
Miscellaneous: SOA, Agile, Git, Anaconda

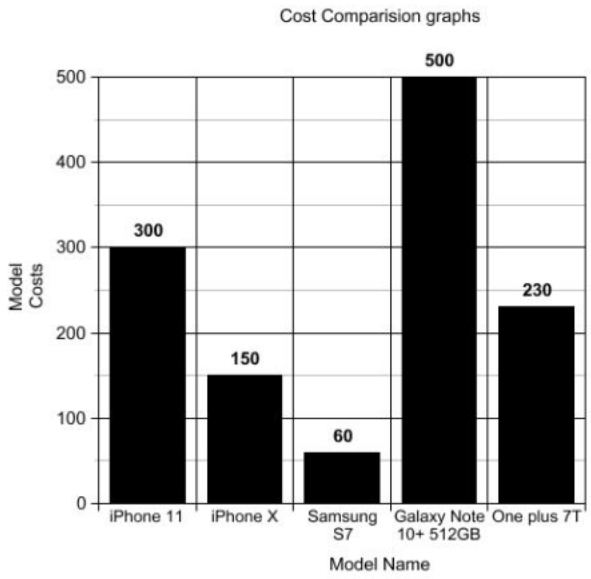
B.Results

The predicted results for various models of phones are depicted based on the consideration of the user inputs and the model outputs.

The initial graph indicates the prices for the phones which are perfect



The below graph indicates the prices of the phone when taken into consideration the model results.



IV. Conclusion

This tool provides ease of access and a fair deal of assessment for the users. The advantages of using these models are

- Provides fair pricing based on machine learning algorithms
- One place stop to get all your needs done. No need to visit multiple shops to check on phone exchanges.
- Instant cash quote display.

Improvements that can be done for the model could be additional model implementation. There are a number of phones being added to the market daily. There should be an implementation to update the models that are being released into the market periodically.

Acknowledgment

The authors gratefully acknowledge the contributions of Jitvans available on https://www.kaggle.com/jitvans for providing the guidelines on crack detection. Also, we would like to thank Rakesh Ranjan, a lecture in San Jose and a full-time employee at IBM. He has provided me the golden opportunity to do this wonderful project on this topic which also helped me in doing a lot of research and we came across many new things and enrich our knowledge and understanding.

References

[1] VGG16 – Convolutional Network for Classification and Detection by Muneeb ul Hassan

[2] A Practical Guide to ReLU - Start using and understanding ReLU without BS or fancy equations. By Danqing Liu

[3] Understand the Softmax Function in Minutes - Uniqtech by UniqtechI.

[4] Goodfellow, Bengio & Courville 2016, p. 184.

[5] Bishop, Christopher M. (2006). Pattern Recognition and Machine Learning. Springe