Mobile Phone Classification using CNN

Computer Vision with CNNs

Domain

Mobile Phones, Computer Vision

Business Context

The ability to process visual information using machine learning algorithms can be very useful. Electronics companies can use it to identify the presence of a mobile in an image (location), mobile brand just by looking at the image, and minor damages if there are any. The computer vision field has multiple applications and based on the available data it can be used to meet business objectives.

Here, we will use a simple convolutional neural network to classify images with and without mobile phones.

Objective

Given the images and label whether an image has a mobile phone or not, can you train a model that allows you to map and find the presence of mobile phones within the selected images.

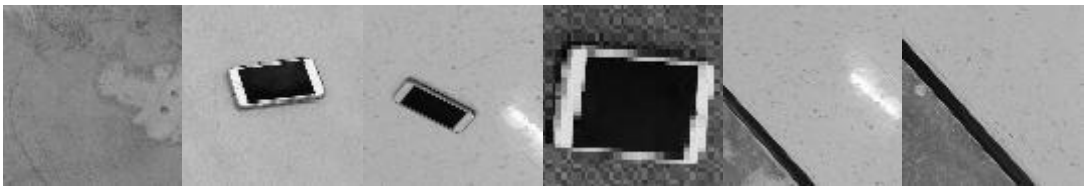
Dataset description

We have two folders inside the “mobile-images” folder.

yes-mobile - mobile phone is available in the image. It has 124 images

no-mobile - mobile phone is not there. It has 121 images

Sample images -



Steps

The objective of the project is to learn how to implement a simple convolutional neural network and understand the basics of image classification.

- Read the data from the folders containing images. Folder name - “mobile-images”.
- Check the number of images with a mobile and without mobile. And report the total number of samples.
- Visualize a few images to understand the data.
- Store the images into an array and link its label. Yes - 1 and No - 0
- Check shape and size of the data
- Prepare independent and dependent variables by combining the above images into one variable and shuffling them.
- Split the data into train and test sets
- Reshape the data to make it suitable for the input layer and normalize it
- One hot encode the labels for train and test data
- Define the model architecture using TensorFlow with a Conv2D layer followed by dense layers with activation as ReLu and softmax respectively.
- Compile the model with loss as categorical cross-entropy and sgd optimizers. Use accuracy as the metric for evaluation
- Fit and evaluate the model. Print the loss and accuracy for the test data
- Plot the graph for training loss and training accuracy change with number of epochs

Further Questions (Optional) -

Can we solve this use case using neural networks?

Do you think the CNNs perform better in comparison to classic ML, NN algorithms in the case of image data?

How can this use case be useful in your organization?

Learning Outcomes

- Computer Vision
- Convolution Neural Networks
- Working with images
- Classification using Convolutional Neural Networks