Portrait Mode of an Image using Convolutional Neural Networks

ITCS 6156

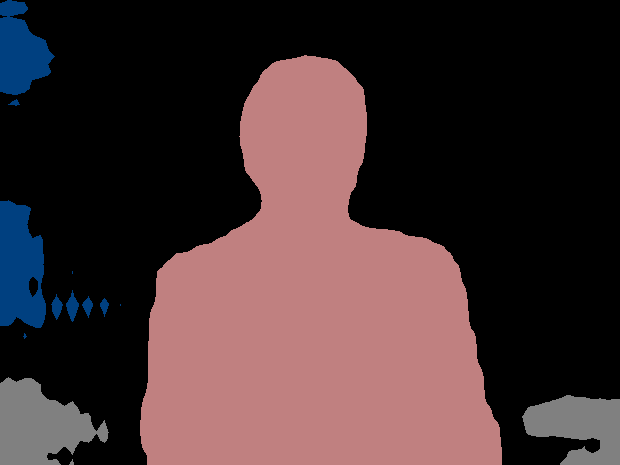
Machine Learning

Final Project Report

(<https://drive.google.com/drive/folders/1vX84SPL7sUgZnhBCztT5K3Mqt__a7OiI?usp=sharing>)

All the folder structure, readme and python files are placed in a zip file placed in the above link

A person wearing a suit and tie

Description generated with very high confidenceA person standing in front of a window posing for the camera

Description generated with very high confidence

**Original Image** **Segmented Image** **Output Portrait Image**

(***Note****:* The above images are the output of our model)

By:

Sri Vishnu Yandrapati

Kedarnath Kurnool Gandla

Sridhar Ramesh Babu

Introduction

Problem Statement

This problem is regarding creating a portrait mode of an image using machine learning techniques without requiring a dual camera setup (an additional hardware) which is usually required to take shallow-depth-field images. This methodology is currently being used in Google’s latest flagship phones Pixel 2 and Pixel 2XL.

A single-lens reflex (SLR) camera with a big lens has a shallow depth of field, meaning that objects at a particular distance from the camera are sharp, while objects in front of or behind that "in-focus plane" are blurry. Shallow depth of field is a good way to draw the viewer's attention to a subject, or to suppress a cluttered background. Shallow depth of field is what gives portraits captured using SLRs their characteristic artistic look.

We will be using Convolution Neural Networks for detecting the foreground and background of an image by enhancing the existing VGG16 standard architecture.

Motivation

Portrait mode is a current trend in latest smart phones which create a depth-of-field effect in an image. For this, the smart phone requires a dual lens camera which is an additional hardware used to create this effect.

Google is the first company that uses Machine Learning techniques which creates this effect without using the dual lens in the camera. It uses foreground and background detection using a trained CNN.

This idea was really a great motivation for us to choose this project, which showcases the widespread usage of machine learning techniques across different areas/industries.

Review of other researches

Convolution Neural Network based systems are increasingly used in many number of areas and plays an important role in the image-wide prediction problems such as segmentation, restoration, reconstruction, flow, etc. Some of the on-going researches mentioned below:

1. Currently in autonomous vehicles, the object detection system is being developed using CNN which detects the object and classifies each pixel based on the background and foreground (semantic segmentation). The algorithms are trained for detecting and classifying the predefined set of object types.
2. Traffic systems are nowadays implementing the latest machine learning techniques to detect the number of vehicles passing through. For this, the video footage is taken and being divided into frames and fed to the model that is being trained using CNN architecture which identifies the vehicle using foreground and background detection.
3. There is also an ongoing research in the medical field where models based on FCN being applied successfully to various 2D/3D medical image segmentation problems (e.g. U-Net).
4. Google had trained a neural network, written in TensorFlow, that looks at the picture, and produces an estimate of which pixels are people and which aren't. The specific network we use is a convolutional neural network (CNN) with skip connections. This is used in their latest flagship phones Pixel 2 and Pixel 2XL.

Open questions in the domain

* Convolution neural network require a huge dataset for training purpose. What if the dataset size does not meet the training requirement?
* A convolution is a significantly slower operation than, say maxpool. If the network is deep, each training step is going to take much longer. How does we overcome this if we need to consider more number of classes and large datasets.
* How does CNN work if the position and orientation of the object change? Will the model still classifies the object in to its category?

Short summary of your proposed approach

To solve this problem, we first need to consider the image segmentation. This is a computer vision task which includes both segmentation and image classification. Segmentation falls into the classification task where each pixel of the image is classified into a class. The main advantage of segmentation will not only help us to detect the object in the image, but also help us to identify the location of the object using the pixel data.

In our project, we use one of the predefined CNN architecture which is available in the Keras library. This library has five pretrained CNN models which are basically trained on ImageNet dataset. VGG network is based on 3x3 convolution layers stacked on top of each other in increasing depth. Two fully-connected layers, each with 4,096 nodes are then followed by a softmax classifier.

In our architecture we have skipped the two MaxPool layers and added dilated convolution layer instead. We have also replaced the fully connected layer with a convolution layer. We have added a dense layer before the convolution layer at end for the best connectivity. All these changes to the original architecture helped us to obtain the best accuracy for the considered dataset.

It involves detecting foreground and background in an image and blur the background. Main challenge is to recognize the foreground irrespective of shape, color, lighting, noise etc. Later, the background of the image will be blurred to give portrait mode.

How reflect instructor’s feedback:

In this report we are explaining Google’s architecture in detail and comparing Google’s model with our model and checking the experimental results.

Backgrounds

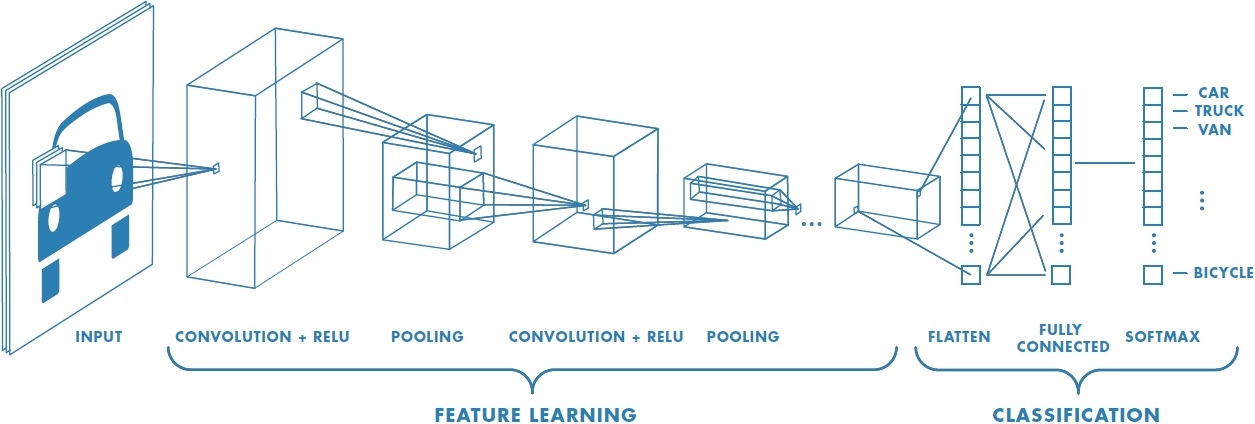
Survey of other related researches

One of the related research deals with generating better results in semantic image segmentation using Atrous Convolution layer and Fully Connected CRFs. In DeepLab-v1, Atrous Convolution Layer has been used to enlarge the field of view of filters to incorporate larger context. This doesn’t involve increase in number of parameters and amount of computation. In DeepLab-v2, an Atrous Spatial Pyramid Pooling (ASPP) layer is included for segmentation of objects at multiple scales. ASPP captures image context and objects effectively by probing incoming convolutional feature layer with filters at multiple sampling rates.

Localization of object boundaries is being improved by combining methods from Deep Convolutional Neural Networks and probabilistic graphical models. Responses at the final DCNN layer are combined with a fully connected Conditional Random Field (CRF) to enhance localization performance. CRFs smoothens the segmentation based on underlying image intensities. In DeepLab-v3, ASPP module is augmented to capture longer range information. Batch Normalization parameters have been included to facilitate training. In DeepLab-v3+, a decoder module is being used to refine segmentation results along object boundaries. SegNet and RefineNet models also includes this Deep Convolutional Encoder-Decoder Architecture.

Fully Convolutional Network (FCN) is one of the initial models which used deconvolutional layers instead of simple bilinear interpolation. Later, PSPNet (Pyramid Scene Parsing Network) model came up which aggregates the context by applying large kernel pooling layers in pyramid pooling module. An auxiliary loss is also applied after fourth stage of ResNet in addition to the loss on main branch. Whereas Large Kernel Matters model improves the semantic segmentation by using Global Convolutional Network. Here, object boundaries are much precisely calculated by making use of residual based boundary refinement.

*Standard deep learning model for image recognition*



Method:

The model we used for this project is VGG16. VGG16 is one of the best models available for image classification. But this model can also be used for image segmentation. Following diagram is the classic VGG16 model. This model is built by Visual Geometry Group in the year 2012. There was lot of modification to this since then.



**Description of layers:**

1. Convolution using 64 filters
2. Convolution using 64 filters + Max pooling
3. Convolution using 128 filters
4. Convolution using 128 filters + Max pooling
5. Convolution using 256 filters
6. Convolution using 256 filters
7. Convolution using 256 filters + Max pooling
8. Convolution using 512 filters
9. Convolution using 512 filters
10. Convolution using 512 filters + Max pooling
11. Convolution using 512 filters
12. Convolution using 512 filters
13. Convolution using 512 filters + Max pooling
14. Fully connected with 4096 nodes
15. Fully connected with 4096 nodes
16. Output layer with Softmax activation with 1000 nodes.

Enhancing VGG16 model in our project

Basic VGG16 model is basically used for classification. Basic VGG16 model was modified. We took a model which has following changes to the existing VGG16 model. Layers from 11 to 16 modified from the existing model.

1. Atrous Convolution using 512 filters.
2. Atrous Convolution using 512 filters.
3. Atrous Convolution using 512 filters + Max pooling
4. Atrous Convolution using 4096 filters.
5. Convolution using 4096 filters.
6. Convolution using 21 filters with activation linear.

In the basic model, there is a repeated combination of max-pooling and downsampling performed by convolutional layers. This type of architecture is good for classification but not segmentation. In order to overcome this problem, we can remove the last few downsampling operation with some upsampling using filters. This way of upsampling the model using filters is done by atrous convolution. To replace a fully connected layer, a combination of atrous convolution followed by bilinear interpolation of features to actual image size. My performing this process, the accuracy of performing image segmentation has increased a lot. The model was created to predict 21 different classes in an image. Object of different class is highlighted in different colors in the final output.

The actual output of the model is the image which has pixels values from 0 to 20. The model output is then converted to final output image of size equal to the input image, but the images are colored according the classes defined in the model.

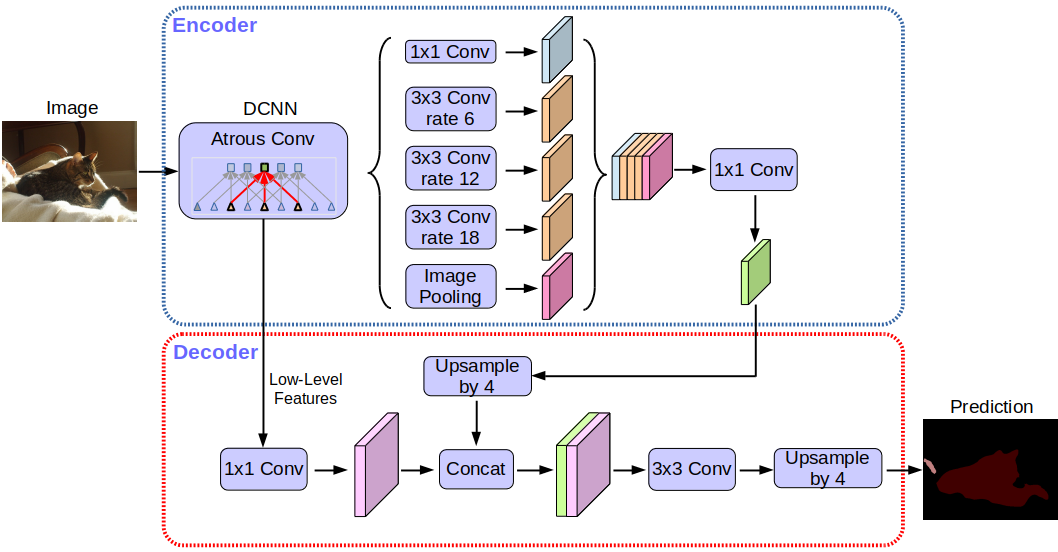
21 predictable classes are mentioned below:

1. Class 0: Background
2. Class 1: Airplane
3. Class 2: Bicycle
4. Class 3: Bird
5. Class 4: Boat
6. Class 5: Bottle
7. Class 6: Bus
8. Class 7: Car
9. Class 8: Cat
10. Class 9: Chair
11. Class 10: Cow
12. Class 11: Dining Table
13. Class 12: Dog
14. Class 13: Horse
15. Class 14: Motor bike
16. Class 15: Person
17. Class 16: Potted Plant
18. Class 17: Sheep
19. Class 18: Sofa
20. Class 19: Train
21. Class 20: TV/ Monitor

To this model, we want to add an additional Fully Connected Layer which will give both the advantage of Dense Layer and the Atrous Convolutional Layer. We added the dense layer after Atrous Convolutional Layer (4096 Filters), as it increases the prediction capacity of various classes in the input image. It uses all the features extracted from previous layers to predict the model. By performing this method, we were able to increase the prediction accuracy for the input model. After performing trial and error method of adding dense layer in-between various layers, we came up with idea.

Comparison Between Model used by Google and modelimplemented in this project

Google has DeepLab-V3+ model for identifying foreground and background in images.



Google’s architecture

We can see from the model diagram that Google has implemented Encoder and Decoder for performing downsampling and upsampling of image. Here Encoding is used for downsampling at the same it used in extracting various features available in the images. After downsampling, we need to should predict the objects in the image. This is done using Atrous Convolution. Then, we need to perform upsampling using decoder.

There are two differences between Google’s implementation and ours.

1. We are not performing Encoding and Decoding. As decoder uses some inputs from downsampling for performing Upsampling. But we have skipped this step. We are performing upsampling without having input from upsampling.
2. We have added an additional dense layer after Atrous Convolution layer.

Experiments

The entire process of creating a portrait mode of an image takes two steps.

1. First generating the segmented image using the enhanced VGG16 architecture. This architecture classifies the images into the desired 21 classifications. Each classification has its own RGB color using which the humans are identified.
2. The second step includes providing the blur effect to the image using the generated segmented image. We compare both segmented image and original image for making the background pixels soften which creates the portrait image

Initially, we tried using different architectures but, the training dataset needs to be very large. Finally, we have decided to enhance the existing VGG16 architecture and got an accuracy of 95%.

We can experiment on different images to check the accuracy of the model generated. As discussed in the above section, the images are classified into 21 classifications.

Now, let us experiment passing the images that contain above objects and see whether we are getting the actual segmented image as expected.

* The below input image illustrates the background, human and bike combination in an image. When we feed this image to our model, we get the following results.

A close up of a logo

Description generated with high confidence

A person riding on the back of a motorcycle

Description generated with high confidenceA person riding a motorcycle on a track

Description generated with very high confidence

Original image segmented image output blur image

* The next image which we consider will be of train. Below are the outputs generated by our model

A picture containing nature

Description generated with very high confidenceA large long train on a steel track

Description generated with very high confidenceA large long train on a steel track

Description generated with very high confidence

Original train image Segmented image generated Blur image generated

* The below image illustrates a horse with a background. The results obtained are below.

A close up of an animal

Description generated with high confidenceA brown horse

Description generated with very high confidenceA brown horse standing in the grass

Description generated with very high confidence

Original Image Segmented Image Output blur image

* The below image illustrates two humans sitting side by side and the outputs are below

A person that is standing in the grass

Description generated with high confidenceA close up of a mountain

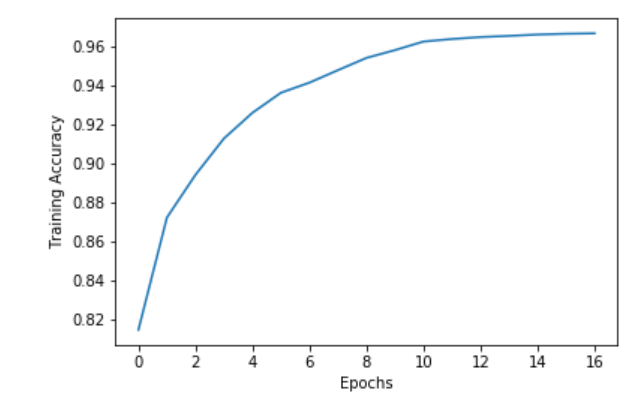
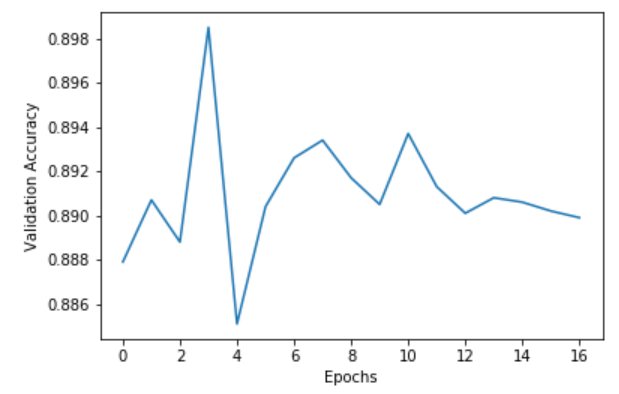
Description generated with high confidenceA person that is sitting on the ground

Description generated with high confidence

Original Image Segmented Image Output blur image

All the above combinations of images are according to the classifications which we have considered as predefined classifications. Our model needs to be trained with various other datasets in order to get more accuracy and with other classifications. We have trained our model on the Benchmark dataset.

Graphs for training and Validation are given below:



Training accuracy vs Epochs Validation accuracy vs Epochs

Conclusions

Overall this project helped us to understand the major researches going on in this field and how machine learning techniques are used across various fields. In our project, we actually learnt how the deep neural networks actually work and came across various CNN architectures and its respective usage in the current on-going research.

Additionally, we learnt how images are actually processed, converted into matrices/arrays and then convoluted further to train the model. Got familiar with various libraries such as tensorflow, keras, os, numpy, etc. which helped us a lot to perform various tasks related to training, validation and preprocessing of the dataset.

We tried to replicate the Google’s model of creating the portrait effect for an image, but we considered a different architecture which gets trained with minimum input training sample. Our model is limited to only 21 classes which are discussed in the above section and can only create blur effect for those images which fall under those 21 classes.

Response to the feedback

We have described the Google’s architecture in detail and also compared the difference in the architecture which we have followed.

References

<https://research.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html?m=1>

<https://towardsdatascience.com/background-removal-with-deep-learning-c4f2104b3157>

<https://arxiv.org/abs/1703.02719>

<http://blog.qure.ai/notes/semantic-segmentation-deep-learning-review>

<https://research.googleblog.com/2018/03/semantic-image-segmentation-with.html>

<https://www.cs.toronto.edu/~frossard/post/vgg16/>

<https://towardsdatascience.com/background-removal-with-deep-learning-c4f2104b3157>