Computer vision and deep learning project

Introduction

Project idea:

Removing of pooling layers in pretrained geolocation models

Motivation: Improving environment detection

Key terminology:

Adjusted model – VGG16\_PLACES365

Pooling layer - Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map.

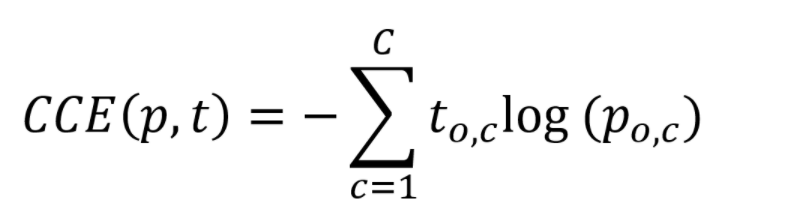
Stride - Stride is a parameter of the neural network's filter that modifies the amount of movement over the image or video.

Loss function -> sparse categorical crossentropy

Sparse categorical crossentropy vs categorical cossentropy

-categorical\_crossentropy (cce) produces a one-hot array containing the probable match for each category,

-sparse\_categorical\_crossentropy (scce) produces a category index of the most likely matching category.



Related work

Most architectures approach down sampling using either average pooling or derivations. In newer architectures, data scientists have removed pooling layers, replacing them with a bigger stride in order to reduce the processing time and avoid losing details.

Reference: <https://github.com/CSAILVision/places365>

Another method for solving this problem is a basic evolutionary algorithm, which has the downside of being incredibly slow and highly innacurate.

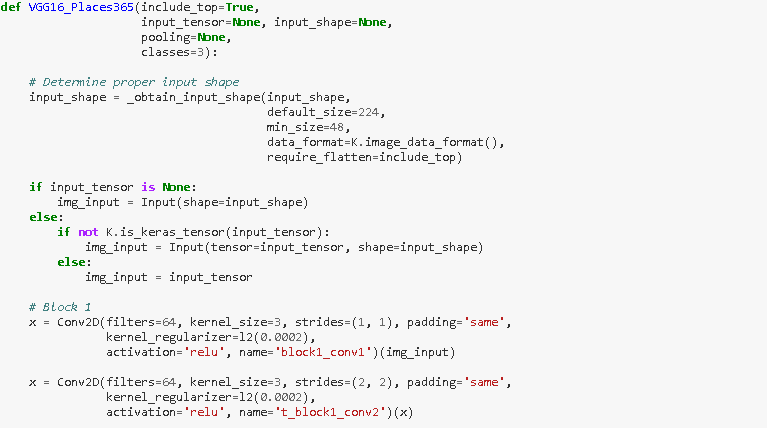
Proposed approach

We’ve decided to remove the pooling layer form the equation and replace it with a bigger stride.

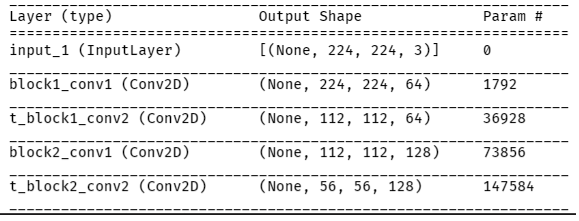
Our architecture is based on the VGG16\_PLACES365 model.

There are 5 convolutional blocks each having their pooling layer replaced. We are using the Adam optimizer and sparse categorical crossentropy loss function for compiling the model.

Input + block 1 + block 2:

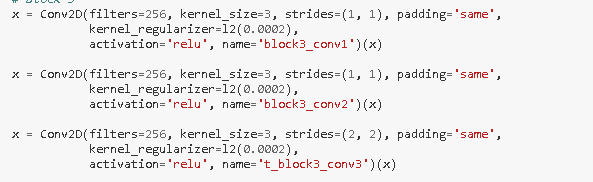


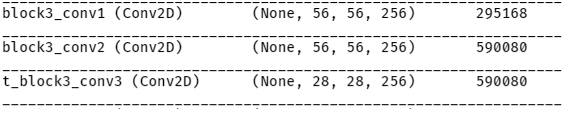




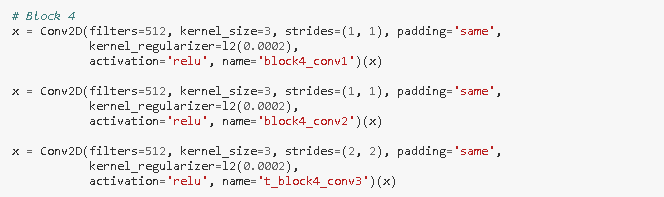
There is 1 input layer and 2 convolutional layers for the first 2 blocks.

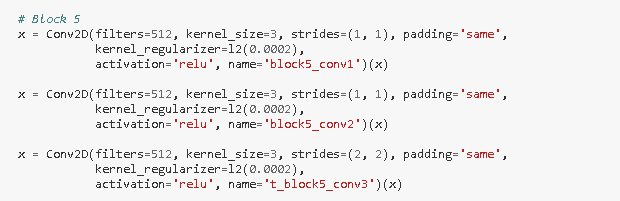
Block 3:

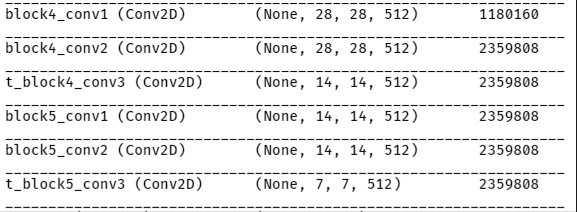




Block 4 + Block 5:





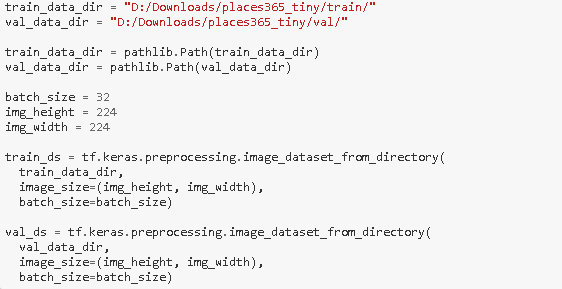


For the Blocks 3 4 and 5 there are 3 convolutional layers.

Experimental results

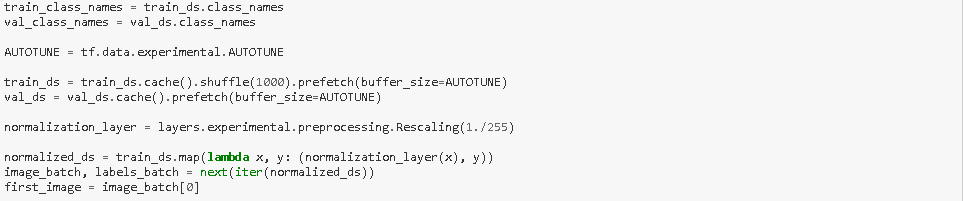
We’ve tried transferring the weights pretrained on the initial model, but this failed to return better results than simply retraining the model form the beginning.

Training initialization:



We initialize the training and validation data and begin preprocessing. For simplicity, we rescale the images to a more manageable size: 244 by 244. The ratio between training and validation is 5:1

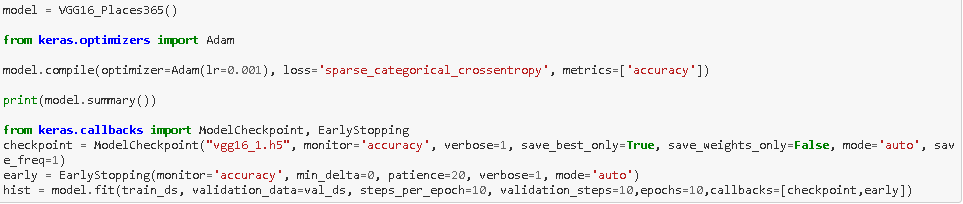
For this experiment’s sake, we’ve only used 10 output classes instead of all of them to test the new model.



We normalize the data set and shuffle them to avoid overtraining for certain categories.

We use the loss and accuracy on the training set and the validation loss and the validation accuracy on the validation set in order to analyze the performance of the model.

a



We initialize the model, with the Adam optimizer, the sparse categorical entropy loss function and we measure the accuracy as certain checkpoints. Then, we start fitting the model with the train data set and validation data.

Final