CO468: Computer Vision Assignment

Team:
Polkampally Keerthana 16CO131
Sharanya Kamath 16CO140
Archana Priyadarshani 16CO207

Submitted to:
Dr. Jeny Rajan
Assistant Professor,
CSE Dept, NITK

Date: May 5th 2020

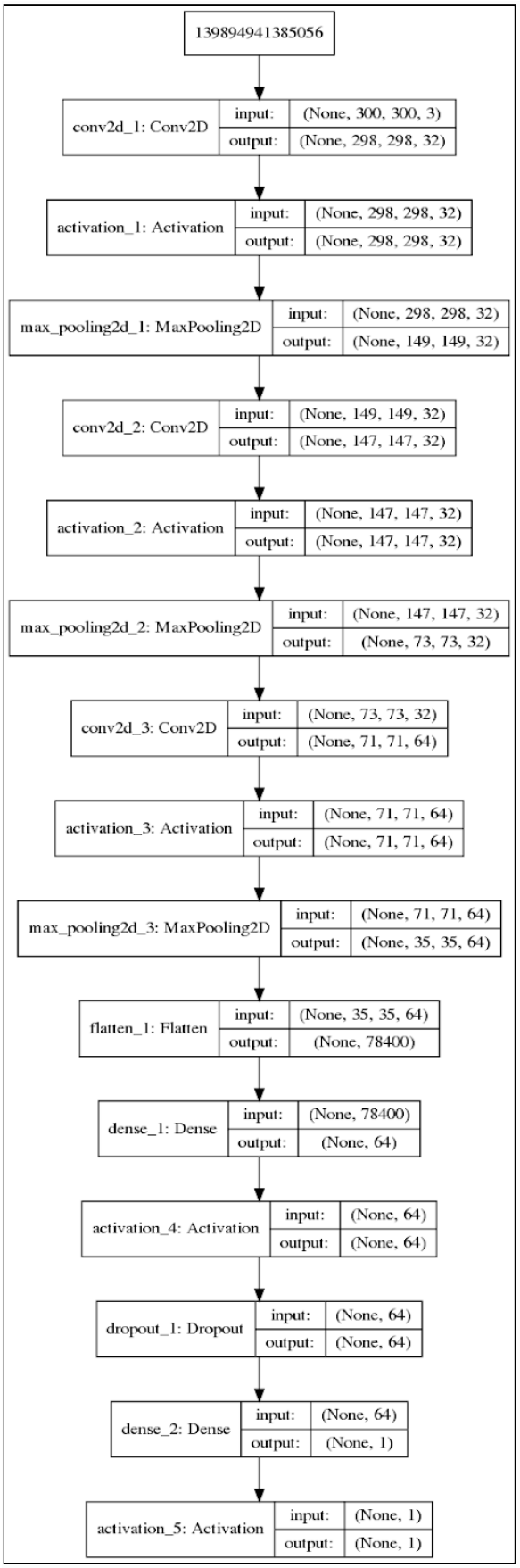
- ## Model Architecture

  The model consists of a stack of 3 convolution layers with a ReLU activation and followed by max-pooling layers. On top of these layers, there are two fully-connected layers. The model ends with a single unit and a sigmoid activation.

  - A benefit of using *convolutional layers* is that the number of parameters to estimate is much lower compared to having a normal hidden layer.
  - The *ReLu layer* adds nonlinearity to the network.
  - *Pooling* aggregates the input volume in order to reduce the dimensions further to speed up computation time as the number of parameters to be estimated is reduced. Also, it helps to avoid overfitting by making the network more robust.
  - The output of the last ReLu layer is flattened to a vector which is then connected to all neurons in the *fully connected layer*.
  - *Dropout* is used to avoid overfitting as it randomly sets a specified fraction of the inputs to zero, during the training of the model.
  - *binary_crossentropy* is used as the loss parameter which is good for binary targets. *Accuracy*, *precision* and *recall* are used as the evaluation metric.

  Data Augmentation is done to avoid overfitting. Data augmentation is done on the training samples via some random transformations like rotation, width and height shift, horizontal flip, flip mode etc. Augmentation on the data is done in batches.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 298, 298, 32)      896
_________________________________________________________________
activation_1 (Activation)    (None, 298, 298, 32)      0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 149, 149, 32)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 147, 147, 32)      9248
_________________________________________________________________
activation_2 (Activation)    (None, 147, 147, 32)      0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 73, 73, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 71, 71, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 71, 71, 64)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 35, 35, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 78400)             0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                5017664
_________________________________________________________________
activation_4 (Activation)    (None, 64)                0
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 65
_________________________________________________________________
activation_5 (Activation)    (None, 1)                 0
=================================================================
Total params: 5,046,369
Trainable params: 5,046,369
Non-trainable params: 0
```


139894941385056
conv2d_1: Conv2D
input: (None, 300, 300, 3)
output: (None, 298, 298, 32)
activation_1: Activation
input: (None, 298, 298, 32)
output: (None, 298, 298, 32)
max_pooling2d_1: MaxPooling2D
input: (None, 298, 298, 32)
output: (None, 149, 149, 32)
conv2d_2: Conv2D
input: (None, 149, 149, 32)
output: (None, 147, 147, 32)
activation_2: Activation
input: (None, 147, 147, 32)
output: (None, 147, 147, 32)
max_pooling2d_2: MaxPooling2D
input: (None, 147, 147, 32)
output: (None, 73, 73, 32)
conv2d_3: Conv2D
input: (None, 73, 73, 32)
output: (None, 71, 71, 64)
activation_3: Activation
input: (None, 71, 71, 64)
output: (None, 71, 71, 64)
max_pooling2d_3: MaxPooling2D
input: (None, 71, 71, 64)
output: (None, 35, 35, 64)
flatten_1: Flatten
input: (None, 35, 35, 64)
output: (None, 78400)
dense_1: Dense
input: (None, 78400)
output: (None, 64)
activation_4: Activation
input: (None, 64)
output: (None, 64)
dropout_1: Dropout
input: (None, 64)
output: (None, 64)
dense_2: Dense
input: (None, 64)
output: (None, 1)
activation_5: Activation
input: (None, 1)
output: (None, 1)

- **Data used**

  Augmented PCB Dataset
  - Defect: 2520 samples
  - Non-Defect: 2520 samples
  - Total: 5040 samples

  Train-Test-Val split:
  - Train: 1512 Defect, 1512 Non-Defect
  - Test: 630 Defect, 630 Non-Defect
  - Val: 378 Defect, 378 Non-Defect

- **Results**

  Test Accuracy: 85.47%
  Precision: 0.88
  Recall: 0.82

  Defective Accuracy = TN / (TN + FN) * 100%
  = **89.05%**
  Non-Defective Accuracy = TP / (TP + FP) * 100%
  = **81.9%**

```
Confusion Matrix
[[561  69]
 [114 516]]
Classification Report
              precision    recall  f1-score   support

      Defect       0.83      0.89      0.86       630
   NoDefect        0.88      0.82      0.85       630

   micro avg       0.85      0.85      0.85      1260
   macro avg       0.86      0.85      0.85      1260
weighted avg       0.86      0.85      0.85      1260
```