DATA.ML.300 Computer Vision
Exercise Round 3

Daniel Kusnetsoff

Task 2.

1.The targets are presented in the form: frame,xmin,xmax,ymin,ymax,class_id

The difference between training set and validation set.

Training set: Training set is filled with samples that are used to fit model. The training set is usually the biggest in size of all sets. Simplified the training set is used to learn the parameters for the model. In supervised learning the training set should include both the features and labels needed for the task.

Validation set: A validation dataset is filled with samples of data that the training set has not seen. -> Used to evaluate the fit of the model (training set) unbiasly and hence tune model hyperparameters . By tuning the hyperparameters we can find the optimal values and create the most effective model.

2. There are 7 convolutional layers that are built of Conv2, Batchnormalization, ELU and MaxPooling. Last layer does not have maxpooling.

Here are the layers.

```
conv1 = Conv2D(32, (5, 5), strides=(1, 1), padding="same", kernel_initializer='he_normal', kernel_regularizer=l2(l2_reg), name='conv1')(x1)

   conv1 = BatchNormalization(axis=3, momentum=0.99, name='bn1')(conv1) # Tensorflow uses filter format [filter_height, filter_width, in_channels, out_channels], hence axis = 3

   conv1 = ELU(name='elu1')(conv1)

   pool1 = MaxPooling2D(pool_size=(2, 2), name='pool1')(conv1)


   conv2 = Conv2D(48, (3, 3), strides=(1, 1), padding="same", kernel_initializer='he_normal', kernel_regularizer=l2(l2_reg), name='conv2')(pool1)

   conv2 = BatchNormalization(axis=3, momentum=0.99, name='bn2')(conv2)

   conv2 = ELU(name='elu2')(conv2)

   pool2 = MaxPooling2D(pool_size=(2, 2), name='pool2')(conv2)


   conv3 = Conv2D(64, (3, 3), strides=(1, 1), padding="same", kernel_initializer='he_normal', kernel_regularizer=l2(l2_reg), name='conv3')(pool2)
```

```
    conv3 = BatchNormalization(axis=3, momentum=0.99, name='bn3')(conv3)

    conv3 = ELU(name='elu3')(conv3)

    pool3 = MaxPooling2D(pool_size=(2, 2), name='pool3')(conv3)


    conv4 = Conv2D(64, (3, 3), strides=(1, 1), padding="same", kernel_initializer='he_normal',
kernel_regularizer=l2(l2_reg), name='conv4')(pool3)

    conv4 = BatchNormalization(axis=3, momentum=0.99, name='bn4')(conv4)

    conv4 = ELU(name='elu4')(conv4)

    pool4 = MaxPooling2D(pool_size=(2, 2), name='pool4')(conv4)


    conv5 = Conv2D(48, (3, 3), strides=(1, 1), padding="same", kernel_initializer='he_normal',
kernel_regularizer=l2(l2_reg), name='conv5')(pool4)

    conv5 = BatchNormalization(axis=3, momentum=0.99, name='bn5')(conv5)

    conv5 = ELU(name='elu5')(conv5)

    pool5 = MaxPooling2D(pool_size=(2, 2), name='pool5')(conv5)


    conv6 = Conv2D(48, (3, 3), strides=(1, 1), padding="same", kernel_initializer='he_normal',
kernel_regularizer=l2(l2_reg), name='conv6')(pool5)

    conv6 = BatchNormalization(axis=3, momentum=0.99, name='bn6')(conv6)

    conv6 = ELU(name='elu6')(conv6)

    pool6 = MaxPooling2D(pool_size=(2, 2), name='pool6')(conv6)


    conv7 = Conv2D(32, (3, 3), strides=(1, 1), padding="same", kernel_initializer='he_normal',
kernel_regularizer=l2(l2_reg), name='conv7')(pool6)

    conv7 = BatchNormalization(axis=3, momentum=0.99, name='bn7')(conv7)

    conv7 = ELU(name='elu7')(conv7)
```

Also identity layer can be counted as a layer for the network.

3 . The loss function is a weighted sum between two attributes: localization loss and confidence loss. Localization loss is based on the mismatch between the ground truth box and boundary box of the prediction. This is defined as the smooth L1 loss that has offset to the dimensions of default bounding box.

The confidence loss is based on the idea, that when we get positive match predictions , we can can use the confidence score for the predicted class to penalize the loss. Negative matches are penalized through base class 0, that is for the cases that nothing is detected. The confidence loss is is defined by the softmax loss of the class scores of the class confidences.