

ECE228 Project

May 24, 2021

```
[1]: from tensorflow.keras.layers import Conv2D, BatchNormalization, Activation
import numpy as np
import cv2
import os

def dataset_setup(data_dir='', n_ims=2975, offset_bias=0, img_dim=256):
    """
        Method to import the training data from CityScape and divide into
        ↪ image-label pairs

        Inputs
        --
        data_dir: string
            Location for the data that is being imported
        n_ims: int
            Number of images contained in the folder chosen
        offset_bias: int
            Optionally, skip some images by starting at a position further than 0
        img_dim: int
            Expected image dimension (assuming square images)

        Outputs
        --
        X: list
            Images
        y: list
            Image labels per pixel
    """

    flist = os.listdir(data_dir)
    img0 = cv2.imread(data_dir+flist[0])

    y_dim,x_dim,_ = np.shape(img0)
    X = np.zeros((n_ims,y_dim,int(x_dim/2),3))
    y = np.zeros((n_ims,y_dim,int(x_dim/2),3))
```

```

k = 0
for f in flist[offset_bias:offset_bias+n_ims]:
    X[k] = cv2.imread(data_dir+f)[:,:img_dim]/img_dim
    y[k] = cv2.imread(data_dir+f)[:img_dim:]/img_dim
    k = k+1

return X, y

```

```

[2]: import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

# Directories containing the train and val data
train_files = "./ECE228_project/data/train/"
test_files = "./ECE228_project/data/val/"

# Setup image-label pairs
x, y = dataset_setup(data_dir=train_files, n_ims=2975, offset_bias=0,
    ↪img_dim=256)
x_train, x_val, y_train, y_val = train_test_split(x, y, test_size=0.1,
    ↪random_state=42)
x_test, y_test = dataset_setup(data_dir=test_files, n_ims=500, offset_bias=0,
    ↪img_dim=256)

```

```

[3]: from tensorflow.keras.layers import MaxPooling2D, Dropout, Conv2DTranspose,
    ↪Conv2D, concatenate
from tensorflow.keras.layers import Conv2D, BatchNormalization, Activation
from tensorflow.keras.backend import binary_crossentropy, square
from tensorflow.keras.backend import sum as ksum
from tensorflow.keras import Model, Input

def reconstruction_loss(y_true, y_pred):
    """
    Using binary_crossentropy from Keras for reconstruction loss
    """

    return ksum(binary_crossentropy(y_true, y_pred), axis=-1)

def conv2d_block(input_tensor, n_filters=16, filter_size=3, activation='relu',
    ↪pad='same', batch_norm=True):
    """
    Custom block method to perform consecutive convolutions with optional batch_
    ↪normalization

    Inputs

```

```

--
input_tensor: tensor
    Input image tensor data structure defined within Keras
n_filters: int
    Depth for the convolution layer outputs
filter_size: int
    Dimensions of the filter convolved with the tensor inputs
activation: string
    Activation function for the intermediate layers between convolutions
pad: string
    Determination of if input shape is maintained in convolution
batch_norm: bool
    Flag if batch normalization is used

Outputs
--
x: tensor
    Twice convolved input with optional batch normalization and activation,
↳non-linearities
"""

x = Conv2D(filters=n_filters, kernel_size=(filter_size, filter_size),
          kernel_initializer='he_normal', padding=pad)(input_tensor)
if batch_norm:
    x = BatchNormalization()(x)
x = Activation(activation)(x)

x = Conv2D(filters=n_filters, kernel_size=(filter_size, filter_size),
          kernel_initializer='he_normal', padding=pad)(x)
if batch_norm:
    x = BatchNormalization()(x)
x = Activation('relu')(x)

return x

def UNET(input_shape=(256,256,3), conv_block=conv2d_block, n_filters=32,
↳dropout=0.5, padding='same', batch_norm=True):
    """
    UNET architecture as originally outlined in https://arxiv.org/pdf/1505.04597.pdf with modifications
    to fit different input dimensions.

    Inputs
    --
    input_shape: tuple(int)
        Tuple in 3D corresponding to the dimensions of the input images

```

```

    conv_block: func
        Custom block method to perform consecutive convolutions with optional
↳batch normalization
    n_filters: int
        Number of filters corresponding to depth of input for next layer
    dropout: float
        Dropout percentage hyperparameter to tune overfitting
    padding: string
        Descriptor determining if padding maintain size during convolutions
    batch_norm: bool
        Determines if batch normalization is used

    Outputs
    --
    model: Model
        Returns model architecture without compile
    """

    tensor = Input(shape=input_shape)

    print('Contracting Path')
    c1 = conv_block(tensor, n_filters * 1, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)
    p1 = MaxPooling2D((2, 2))(c1)
    p1 = Dropout(dropout)(p1)

    c2 = conv_block(p1, n_filters * 2, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)
    p2 = MaxPooling2D((2, 2))(c2)
    p2 = Dropout(dropout)(p2)

    c3 = conv_block(p2, n_filters * 4, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)
    p3 = MaxPooling2D((2, 2))(c3)
    p3 = Dropout(dropout)(p3)

    c4 = conv_block(p3, n_filters * 8, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)
    p4 = MaxPooling2D((2, 2))(c4)
    p4 = Dropout(dropout)(p4)

    c5 = conv_block(p4, n_filters * 16, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)

    print('Expanding Path')

```

```

        u6 = Conv2DTranspose(n_filters * 8, (3, 3), strides=(2, 2),
        ↳padding=padding)(c5)
        u6 = concatenate([u6, c4])
        u6 = Dropout(dropout)(u6)
        c6 = conv_block(u6, n_filters * 8, filter_size=3, activation='relu',
        ↳pad=padding, batch_norm=batch_norm)

        u7 = Conv2DTranspose(n_filters * 4, (3, 3), strides=(2, 2),
        ↳padding=padding)(c6)
        u7 = concatenate([u7, c3])
        u7 = Dropout(dropout)(u7)
        c7 = conv_block(u7, n_filters * 4, filter_size=3, activation='relu',
        ↳pad=padding, batch_norm=batch_norm)

        u8 = Conv2DTranspose(n_filters * 2, (3, 3), strides=(2, 2),
        ↳padding=padding)(c7)
        u8 = concatenate([u8, c2])
        u8 = Dropout(dropout)(u8)
        c8 = conv_block(u8, n_filters * 2, filter_size=3, activation='relu',
        ↳pad=padding, batch_norm=batch_norm)

        u9 = Conv2DTranspose(n_filters * 1, (3, 3), strides=(2, 2),
        ↳padding=padding)(c8)
        u9 = concatenate([u9, c1])
        u9 = Dropout(dropout)(u9)
        c9 = conv_block(u9, n_filters * 1, filter_size=3, activation='relu',
        ↳pad=padding, batch_norm=batch_norm)

        outputs = Conv2D(3, (1, 1), activation='sigmoid')(c9)
        model = Model(inputs=[tensor], outputs=[outputs])

        # Return model architecture
        return model

def UNET_plusplus(input_shape=(256,256,3), conv_block=conv2d_block,
↳n_filters=32, dropout=0.5, padding='same', batch_norm=True):
    """
        UNET++ architecture as originally outlined in https://arxiv.org/pdf/1807.10165.pdf
        ↳with modifications to fit different input dimensions.

        Inputs
        --
        input_shape: tuple(int)
            Tuple in 3D corresponding to the dimensions of the input images
    """

```

```

conv_block: func
    Custom block method to perform consecutive convolutions with optional
    ↪ batch normalization
    n_filters: int
        Number of filters corresponding to depth of input for next layer
    dropout: float
        Dropout percentage hyperparameter to tune overfitting
    padding: string
        Descriptor determining if padding maintain size during convolutions
    batch_norm: bool
        Determines if batch normalization is used

    Outputs
    --
    model: Model
        Returns model architecture without compile
    """

tensor = Input(shape=input_shape)

print('Backbone')

c00 = conv_block(tensor, n_filters * 1, filter_size=3, activation='relu',
    ↪ pad=padding, batch_norm=batch_norm)
p00 = MaxPooling2D((2, 2))(c00)
p00 = Dropout(dropout)(p00)

c10 = conv_block(p00, n_filters * 2, filter_size=3, activation='relu',
    ↪ pad=padding, batch_norm=batch_norm)
p10 = MaxPooling2D((2, 2))(c10)
p10 = Dropout(dropout)(p10)

c20 = conv_block(p10, n_filters * 4, filter_size=3, activation='relu',
    ↪ pad=padding, batch_norm=batch_norm)
p20 = MaxPooling2D((2, 2))(c20)
p20 = Dropout(dropout)(p20)

c30 = conv_block(p20, n_filters * 8, filter_size=3, activation='relu',
    ↪ pad=padding, batch_norm=batch_norm)
p30 = MaxPooling2D((2, 2))(c30)
p30 = Dropout(dropout)(p30)

c40 = conv_block(p30, n_filters * 16, filter_size=3, activation='relu',
    ↪ pad=padding, batch_norm=batch_norm)

print('First Up Path')

```

```

    u01 = Conv2DTranspose(n_filters * 1, (3, 3), strides=(2, 2),
↳padding=padding)(c10)
    u01 = concatenate([u01, c00])
    u01 = Dropout(dropout)(u01)
    c01 = conv_block(u01, n_filters * 1, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)

    print('Second Up Path')

    u11 = Conv2DTranspose(n_filters * 2, (3, 3), strides=(2, 2),
↳padding=padding)(c20)
    u11 = concatenate([u11, c10])
    u11 = Dropout(dropout)(u11)
    c11 = conv_block(u11, n_filters * 2, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)

    u02 = Conv2DTranspose(n_filters * 1, (3, 3), strides=(2, 2),
↳padding=padding)(c11)
    u02 = concatenate([u02, c01, c00])
    u02 = Dropout(dropout)(u02)
    c02 = conv_block(u02, n_filters * 1, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)

    print('Third Up Path')

    u21 = Conv2DTranspose(n_filters * 4, (3, 3), strides=(2, 2),
↳padding=padding)(c30)
    u21 = concatenate([u21, c20])
    u21 = Dropout(dropout)(u21)
    c21 = conv_block(u21, n_filters * 4, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)

    u12 = Conv2DTranspose(n_filters * 2, (3, 3), strides=(2, 2),
↳padding=padding)(c21)
    u12 = concatenate([u12, c11, c10])
    u12 = Dropout(dropout)(u12)
    c12 = conv_block(u12, n_filters * 2, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)

    u03 = Conv2DTranspose(n_filters * 1, (3, 3), strides=(2, 2),
↳padding=padding)(c12)
    u03 = concatenate([u03, c02, c01, c00])
    u03 = Dropout(dropout)(u03)
    c03 = conv_block(u03, n_filters * 1, filter_size=3, activation='relu',
↳pad=padding, batch_norm=batch_norm)

```

```

print('Final Up Path')

u31 = Conv2DTranspose(n_filters * 8, (3, 3), strides=(2, 2),
padding=padding)(c40)
u31 = concatenate([u31, c30])
u31 = Dropout(dropout)(u31)
c31 = conv_block(u31, n_filters * 8, filter_size=3, activation='relu',
pad=padding, batch_norm=batch_norm)

u22 = Conv2DTranspose(n_filters * 4, (3, 3), strides=(2, 2),
padding=padding)(c31)
u22 = concatenate([u22, c21, c20])
u22 = Dropout(dropout)(u22)
c22 = conv_block(u22, n_filters * 4, filter_size=3, activation='relu',
pad=padding, batch_norm=batch_norm)

u13 = Conv2DTranspose(n_filters * 2, (3, 3), strides=(2, 2),
padding=padding)(c22)
u13 = concatenate([u13, c12, c11, c10])
u13 = Dropout(dropout)(u13)
c13 = conv_block(u13, n_filters * 2, filter_size=3, activation='relu',
pad=padding, batch_norm=batch_norm)

u04 = Conv2DTranspose(n_filters * 1, (3, 3), strides=(2, 2),
padding=padding)(c13)
u04 = concatenate([u04, c03, c02, c01, c00])
u04 = Dropout(dropout)(u04)
c04 = conv_block(u04, n_filters * 1, filter_size=3, activation='relu',
pad=padding, batch_norm=batch_norm)

#Outputs
outputs = Conv2D(3, (1, 1), activation='sigmoid')(c04)
model = Model(inputs=[tensor], outputs=[outputs])

# Return model architecture
return model

```

```

[4]: # Initialize UNET
model = UNET_plusplus(input_shape=(256,256,3), conv_block=conv2d_block,
n_filters=32, dropout=0.5, padding='same', batch_norm=True)
print(model.summary())

```

Backbone
First Up Path
Second Up Path
Third Up Path

Final Up Path
Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 256, 256, 3) 0		
conv2d (Conv2D)	(None, 256, 256, 32) 896		input_1[0][0]
batch_normalization (BatchNorma	(None, 256, 256, 32) 128		conv2d[0][0]
activation (Activation)	(None, 256, 256, 32) 0		batch_normalization[0][0]
conv2d_1 (Conv2D)	(None, 256, 256, 32) 9248		activation[0][0]
batch_normalization_1 (BatchNor	(None, 256, 256, 32) 128		conv2d_1[0][0]
activation_1 (Activation)	(None, 256, 256, 32) 0		batch_normalization_1[0][0]
max_pooling2d (MaxPooling2D)	(None, 128, 128, 32) 0		activation_1[0][0]
dropout (Dropout)	(None, 128, 128, 32) 0		max_pooling2d[0][0]
conv2d_2 (Conv2D)	(None, 128, 128, 64) 18496		dropout[0][0]
batch_normalization_2 (BatchNor	(None, 128, 128, 64) 256		conv2d_2[0][0]
activation_2 (Activation)	(None, 128, 128, 64) 0		batch_normalization_2[0][0]

```

-----
conv2d_3 (Conv2D)                (None, 128, 128, 64) 36928
activation_2[0][0]

-----

batch_normalization_3 (BatchNor (None, 128, 128, 64) 256      conv2d_3[0][0]
-----

activation_3 (Activation)        (None, 128, 128, 64) 0
batch_normalization_3[0][0]

-----

max_pooling2d_1 (MaxPooling2D)  (None, 64, 64, 64)  0
activation_3[0][0]

-----

dropout_1 (Dropout)              (None, 64, 64, 64)  0
max_pooling2d_1[0][0]

-----

conv2d_4 (Conv2D)                (None, 64, 64, 128) 73856      dropout_1[0][0]
-----

batch_normalization_4 (BatchNor (None, 64, 64, 128) 512      conv2d_4[0][0]
-----

activation_4 (Activation)        (None, 64, 64, 128) 0
batch_normalization_4[0][0]

-----

conv2d_5 (Conv2D)                (None, 64, 64, 128) 147584
activation_4[0][0]

-----

batch_normalization_5 (BatchNor (None, 64, 64, 128) 512      conv2d_5[0][0]
-----

activation_5 (Activation)        (None, 64, 64, 128) 0
batch_normalization_5[0][0]

-----

max_pooling2d_2 (MaxPooling2D)  (None, 32, 32, 128) 0
activation_5[0][0]

-----

dropout_2 (Dropout)              (None, 32, 32, 128) 0
max_pooling2d_2[0][0]
-----

```

```

-----
conv2d_6 (Conv2D)                (None, 32, 32, 256) 295168      dropout_2[0][0]
-----
-----
batch_normalization_6 (BatchNor (None, 32, 32, 256) 1024      conv2d_6[0][0]
-----
-----
activation_6 (Activation)         (None, 32, 32, 256) 0
batch_normalization_6[0][0]
-----
-----
conv2d_7 (Conv2D)                (None, 32, 32, 256) 590080
activation_6[0][0]
-----
-----
batch_normalization_7 (BatchNor (None, 32, 32, 256) 1024      conv2d_7[0][0]
-----
-----
activation_7 (Activation)         (None, 32, 32, 256) 0
batch_normalization_7[0][0]
-----
-----
max_pooling2d_3 (MaxPooling2D)   (None, 16, 16, 256) 0
activation_7[0][0]
-----
-----
dropout_3 (Dropout)              (None, 16, 16, 256) 0
max_pooling2d_3[0][0]
-----
-----
conv2d_8 (Conv2D)                (None, 16, 16, 512) 1180160     dropout_3[0][0]
-----
-----
batch_normalization_8 (BatchNor (None, 16, 16, 512) 2048      conv2d_8[0][0]
-----
-----
activation_8 (Activation)         (None, 16, 16, 512) 0
batch_normalization_8[0][0]
-----
-----
conv2d_9 (Conv2D)                (None, 16, 16, 512) 2359808
activation_8[0][0]
-----
-----
batch_normalization_9 (BatchNor (None, 16, 16, 512) 2048      conv2d_9[0][0]
-----
-----
activation_9 (Activation)         (None, 16, 16, 512) 0

```

```

batch_normalization_9[0][0]
-----

conv2d_transpose_6 (Conv2DTrans (None, 32, 32, 256) 1179904
activation_9[0][0]
-----

concatenate_6 (Concatenate)      (None, 32, 32, 512) 0
conv2d_transpose_6[0][0]
activation_7[0][0]
-----

conv2d_transpose_3 (Conv2DTrans (None, 64, 64, 128) 295040
activation_7[0][0]
-----

dropout_10 (Dropout)             (None, 32, 32, 512) 0
concatenate_6[0][0]
-----

concatenate_3 (Concatenate)      (None, 64, 64, 256) 0
conv2d_transpose_3[0][0]
activation_5[0][0]
-----

conv2d_transpose_1 (Conv2DTrans (None, 128, 128, 64) 73792
activation_5[0][0]
-----

conv2d_22 (Conv2D)               (None, 32, 32, 256) 1179904
dropout_10[0][0]
-----

dropout_7 (Dropout)              (None, 64, 64, 256) 0
concatenate_3[0][0]
-----

concatenate_1 (Concatenate)      (None, 128, 128, 128 0
conv2d_transpose_1[0][0]
activation_3[0][0]
-----

conv2d_transpose (Conv2DTranspo (None, 256, 256, 32) 18464
activation_3[0][0]
-----

batch_normalization_22 (BatchNo (None, 32, 32, 256) 1024      conv2d_22[0][0]
-----

```

```

-----
conv2d_16 (Conv2D)                (None, 64, 64, 128) 295040      dropout_7[0][0]
-----
-----
dropout_5 (Dropout)                (None, 128, 128, 128) 0
concatenate_1[0][0]
-----
-----
concatenate (Concatenate)          (None, 256, 256, 64) 0
conv2d_transpose[0][0]
activation_1[0][0]
-----
-----
activation_22 (Activation)          (None, 32, 32, 256) 0
batch_normalization_22[0][0]
-----
-----
batch_normalization_16 (BatchNo (None, 64, 64, 128) 512      conv2d_16[0][0]
-----
-----
conv2d_12 (Conv2D)                (None, 128, 128, 64) 73792      dropout_5[0][0]
-----
-----
dropout_4 (Dropout)                (None, 256, 256, 64) 0
concatenate[0][0]
-----
-----
conv2d_23 (Conv2D)                (None, 32, 32, 256) 590080
activation_22[0][0]
-----
-----
activation_16 (Activation)          (None, 64, 64, 128) 0
batch_normalization_16[0][0]
-----
-----
batch_normalization_12 (BatchNo (None, 128, 128, 64) 256      conv2d_12[0][0]
-----
-----
conv2d_10 (Conv2D)                (None, 256, 256, 32) 18464      dropout_4[0][0]
-----
-----
batch_normalization_23 (BatchNo (None, 32, 32, 256) 1024      conv2d_23[0][0]
-----
-----
conv2d_17 (Conv2D)                (None, 64, 64, 128) 147584
activation_16[0][0]
-----
-----

```

```

activation_12 (Activation)          (None, 128, 128, 64) 0
batch_normalization_12[0][0]

-----

batch_normalization_10 (BatchNo (None, 256, 256, 32) 128          conv2d_10[0][0]

-----

activation_23 (Activation)          (None, 32, 32, 256) 0
batch_normalization_23[0][0]

-----

batch_normalization_17 (BatchNo (None, 64, 64, 128) 512          conv2d_17[0][0]

-----

conv2d_13 (Conv2D)                  (None, 128, 128, 64) 36928
activation_12[0][0]

-----

activation_10 (Activation)          (None, 256, 256, 32) 0
batch_normalization_10[0][0]

-----

conv2d_transpose_7 (Conv2DTrans (None, 64, 64, 128) 295040
activation_23[0][0]

-----

activation_17 (Activation)          (None, 64, 64, 128) 0
batch_normalization_17[0][0]

-----

batch_normalization_13 (BatchNo (None, 128, 128, 64) 256          conv2d_13[0][0]

-----

conv2d_11 (Conv2D)                  (None, 256, 256, 32) 9248
activation_10[0][0]

-----

concatenate_7 (Concatenate)        (None, 64, 64, 384) 0
conv2d_transpose_7[0][0]
activation_17[0][0]
activation_5[0][0]

-----

conv2d_transpose_4 (Conv2DTrans (None, 128, 128, 64) 73792
activation_17[0][0]

-----

activation_13 (Activation)          (None, 128, 128, 64) 0

```

```

batch_normalization_13[0][0]
-----
batch_normalization_11 (BatchNo (None, 256, 256, 32) 128          conv2d_11[0][0]
-----
dropout_11 (Dropout)          (None, 64, 64, 384) 0
concatenate_7[0][0]
-----
concatenate_4 (Concatenate)    (None, 128, 128, 192 0
conv2d_transpose_4[0][0]
activation_13[0][0]
activation_3[0][0]
-----
conv2d_transpose_2 (Conv2DTrans (None, 256, 256, 32) 18464
activation_13[0][0]
-----
activation_11 (Activation)      (None, 256, 256, 32) 0
batch_normalization_11[0][0]
-----
conv2d_24 (Conv2D)             (None, 64, 64, 128) 442496
dropout_11[0][0]
-----
dropout_8 (Dropout)            (None, 128, 128, 192 0
concatenate_4[0][0]
-----
concatenate_2 (Concatenate)    (None, 256, 256, 96) 0
conv2d_transpose_2[0][0]
activation_11[0][0]
activation_1[0][0]
-----
batch_normalization_24 (BatchNo (None, 64, 64, 128) 512          conv2d_24[0][0]
-----
conv2d_18 (Conv2D)             (None, 128, 128, 64) 110656          dropout_8[0][0]
-----
dropout_6 (Dropout)            (None, 256, 256, 96) 0
concatenate_2[0][0]
-----

```

```

activation_24 (Activation)      (None, 64, 64, 128)  0
batch_normalization_24[0][0]

-----

batch_normalization_18 (BatchNo (None, 128, 128, 64) 256      conv2d_18[0][0]
-----

conv2d_14 (Conv2D)             (None, 256, 256, 32) 27680      dropout_6[0][0]
-----

conv2d_25 (Conv2D)             (None, 64, 64, 128)  147584
activation_24[0][0]

-----

activation_18 (Activation)      (None, 128, 128, 64) 0
batch_normalization_18[0][0]

-----

batch_normalization_14 (BatchNo (None, 256, 256, 32) 128      conv2d_14[0][0]
-----

batch_normalization_25 (BatchNo (None, 64, 64, 128)  512      conv2d_25[0][0]
-----

conv2d_19 (Conv2D)             (None, 128, 128, 64) 36928
activation_18[0][0]

-----

activation_14 (Activation)      (None, 256, 256, 32) 0
batch_normalization_14[0][0]

-----

activation_25 (Activation)      (None, 64, 64, 128)  0
batch_normalization_25[0][0]

-----

batch_normalization_19 (BatchNo (None, 128, 128, 64) 256      conv2d_19[0][0]
-----

conv2d_15 (Conv2D)             (None, 256, 256, 32) 9248
activation_14[0][0]

-----

conv2d_transpose_8 (Conv2DTrans (None, 128, 128, 64) 73792
activation_25[0][0]

-----

activation_19 (Activation)      (None, 128, 128, 64) 0

```



```

batch_normalization_19[0][0]
-----
-----
batch_normalization_15 (BatchNo (None, 256, 256, 32) 128          conv2d_15[0][0]
-----
-----
concatenate_8 (Concatenate)      (None, 128, 128, 256 0
conv2d_transpose_8[0][0]
activation_19[0][0]
activation_13[0][0]
activation_3[0][0]
-----
-----
conv2d_transpose_5 (Conv2DTrans (None, 256, 256, 32) 18464
activation_19[0][0]
-----
-----
activation_15 (Activation)        (None, 256, 256, 32) 0
batch_normalization_15[0][0]
-----
-----
dropout_12 (Dropout)             (None, 128, 128, 256 0
concatenate_8[0][0]
-----
-----
concatenate_5 (Concatenate)      (None, 256, 256, 128 0
conv2d_transpose_5[0][0]
activation_15[0][0]
activation_11[0][0]
activation_1[0][0]
-----
-----
conv2d_26 (Conv2D)               (None, 128, 128, 64) 147520
dropout_12[0][0]
-----
-----
dropout_9 (Dropout)              (None, 256, 256, 128 0
concatenate_5[0][0]
-----
-----
batch_normalization_26 (BatchNo (None, 128, 128, 64) 256          conv2d_26[0][0]
-----
-----
conv2d_20 (Conv2D)               (None, 256, 256, 32) 36896          dropout_9[0][0]
-----
-----
activation_26 (Activation)        (None, 128, 128, 64) 0
batch_normalization_26[0][0]

```

```

-----
batch_normalization_20 (BatchNo (None, 256, 256, 32) 128          conv2d_20[0] [0]
-----

conv2d_27 (Conv2D)          (None, 128, 128, 64) 36928
activation_26[0] [0]
-----

activation_20 (Activation)    (None, 256, 256, 32) 0
batch_normalization_20[0] [0]
-----

batch_normalization_27 (BatchNo (None, 128, 128, 64) 256          conv2d_27[0] [0]
-----

conv2d_21 (Conv2D)          (None, 256, 256, 32) 9248
activation_20[0] [0]
-----

activation_27 (Activation)    (None, 128, 128, 64) 0
batch_normalization_27[0] [0]
-----

batch_normalization_21 (BatchNo (None, 256, 256, 32) 128          conv2d_21[0] [0]
-----

conv2d_transpose_9 (Conv2DTrans (None, 256, 256, 32) 18464
activation_27[0] [0]
-----

activation_21 (Activation)    (None, 256, 256, 32) 0
batch_normalization_21[0] [0]
-----

concatenate_9 (Concatenate)   (None, 256, 256, 160 0
conv2d_transpose_9[0] [0]
activation_21[0] [0]
activation_15[0] [0]
activation_11[0] [0]
activation_1[0] [0]
-----

dropout_13 (Dropout)          (None, 256, 256, 160 0
concatenate_9[0] [0]
-----

conv2d_28 (Conv2D)          (None, 256, 256, 32) 46112

```

```

dropout_13[0][0]

-----
batch_normalization_28 (BatchNormalizer) (None, 256, 256, 32) 128 conv2d_28[0][0]
-----

activation_28 (Activation) (None, 256, 256, 32) 0
batch_normalization_28[0][0]

-----

conv2d_29 (Conv2D) (None, 256, 256, 32) 9248
activation_28[0][0]

-----

batch_normalization_29 (BatchNormalizer) (None, 256, 256, 32) 128 conv2d_29[0][0]
-----

activation_29 (Activation) (None, 256, 256, 32) 0
batch_normalization_29[0][0]

-----

conv2d_30 (Conv2D) (None, 256, 256, 3) 99
activation_29[0][0]
=====
Total params: 10,203,715
Trainable params: 10,196,419
Non-trainable params: 7,296
-----

None

```

```
[5]: # Compile model with specified optimizer and loss
model.compile(optimizer='adam', loss='mse')
```

```
[6]: # Track model history as it trains
import tensorflow as tf
print(tf.__version__)

h = model.fit(x_train, y_train, epochs=5, shuffle=True, batch_size=1,
    ↪ validation_data=(x_val, y_val))
```

2.1.0

Train on 2677 samples, validate on 298 samples

Epoch 1/5

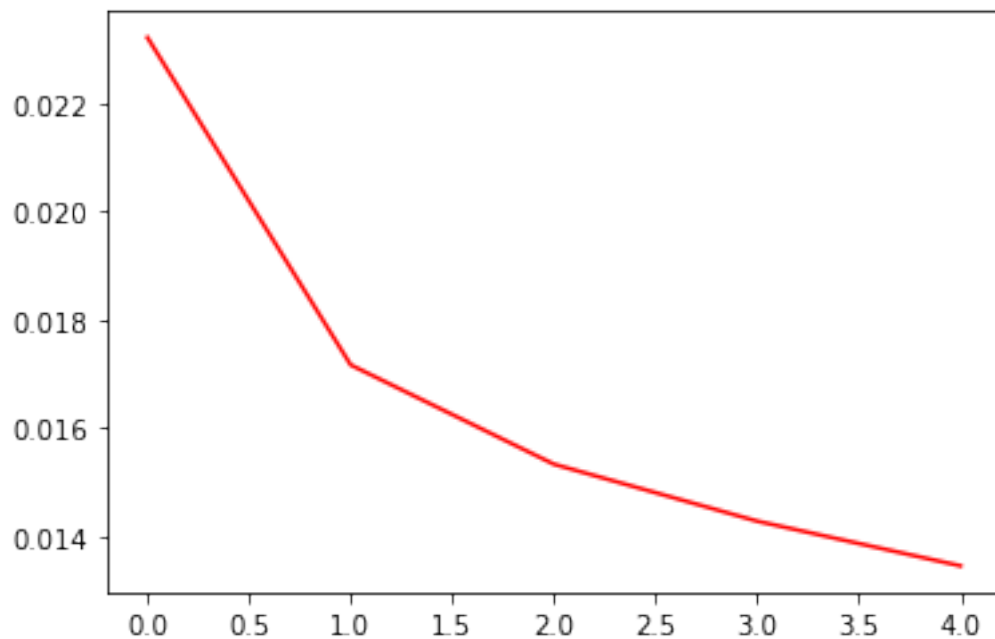
2677/2677 [=====] - 185s 69ms/sample - loss: 0.0232 -

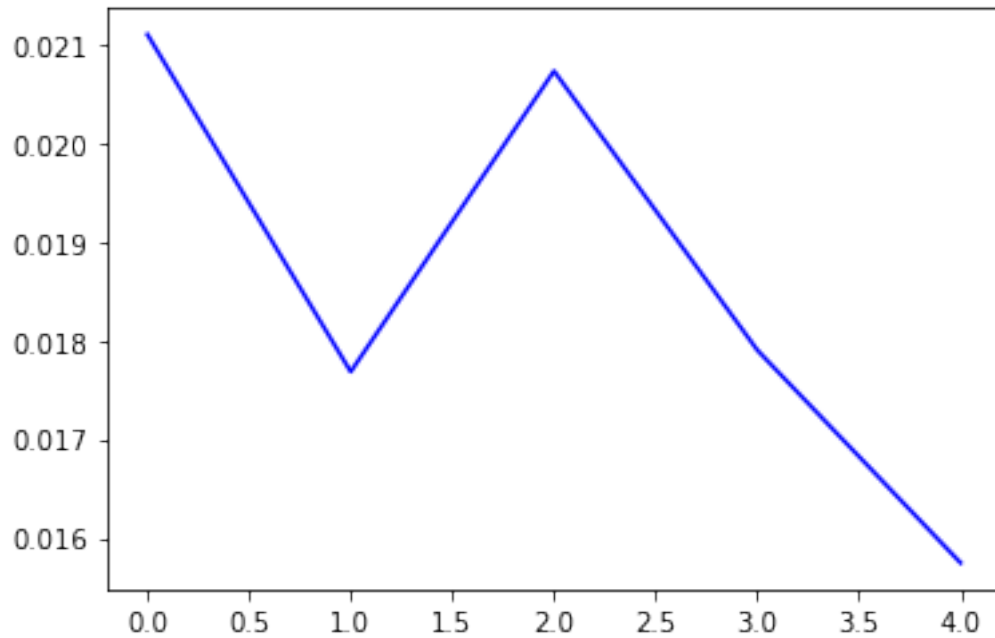
val_loss: 0.0211

Epoch 2/5

```
2677/2677 [=====] - 176s 66ms/sample - loss: 0.0172 -  
val_loss: 0.0177  
Epoch 3/5  
2677/2677 [=====] - 175s 65ms/sample - loss: 0.0153 -  
val_loss: 0.0207  
Epoch 4/5  
2677/2677 [=====] - 175s 65ms/sample - loss: 0.0143 -  
val_loss: 0.0179  
Epoch 5/5  
2677/2677 [=====] - 176s 66ms/sample - loss: 0.0135 -  
val_loss: 0.0157
```

```
[7]: # Print results for training MSE and validation MSE  
plt.plot(h.history['loss'], 'r')  
plt.show()  
plt.plot(h.history['val_loss'], 'b')  
plt.show()
```

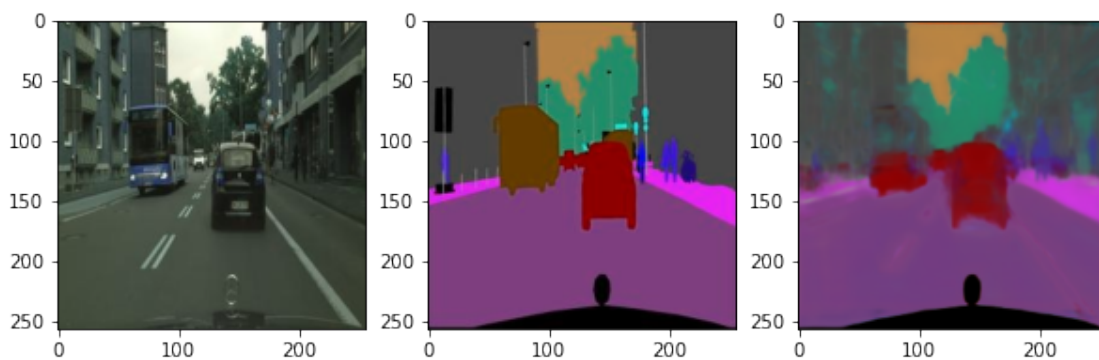


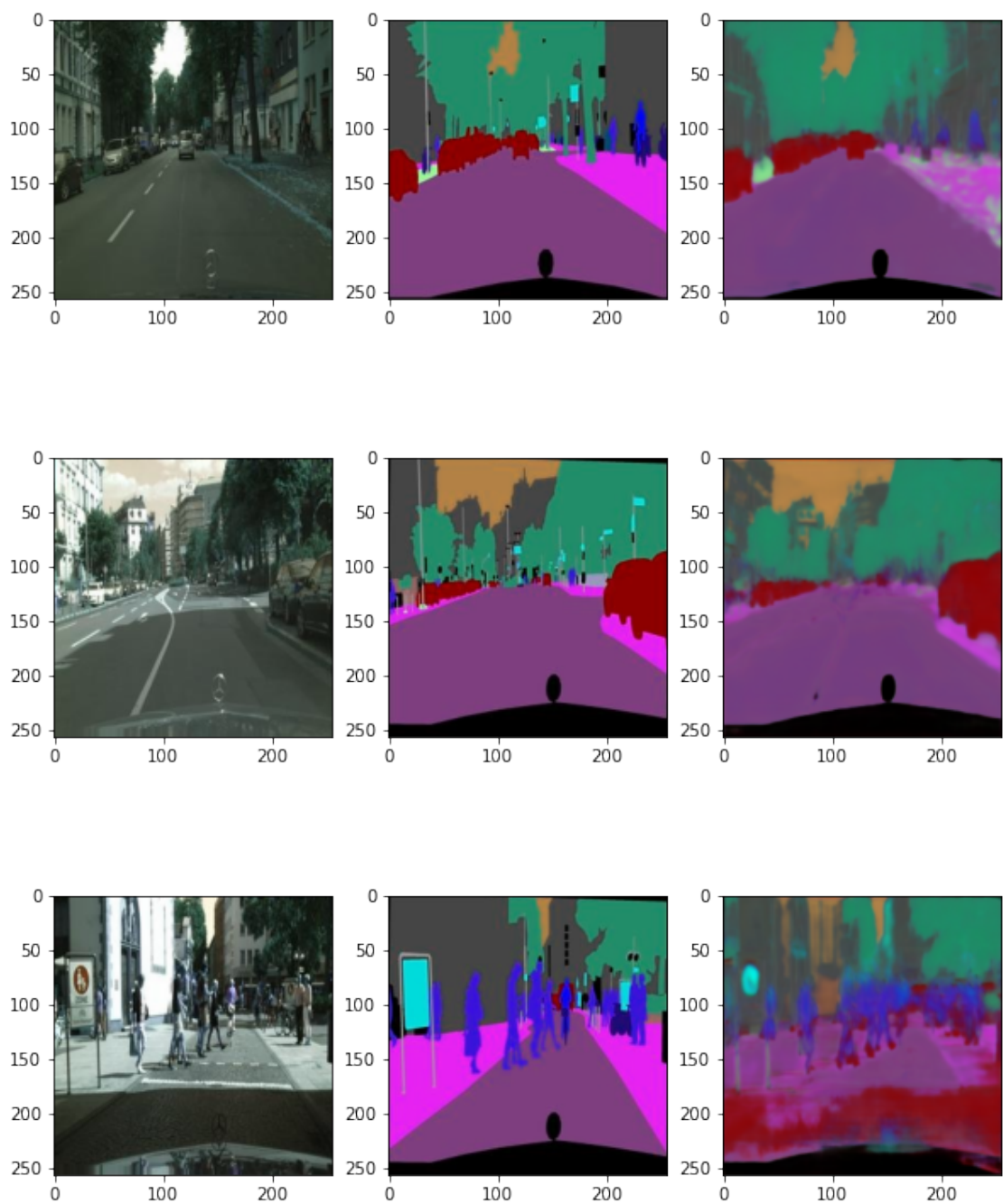


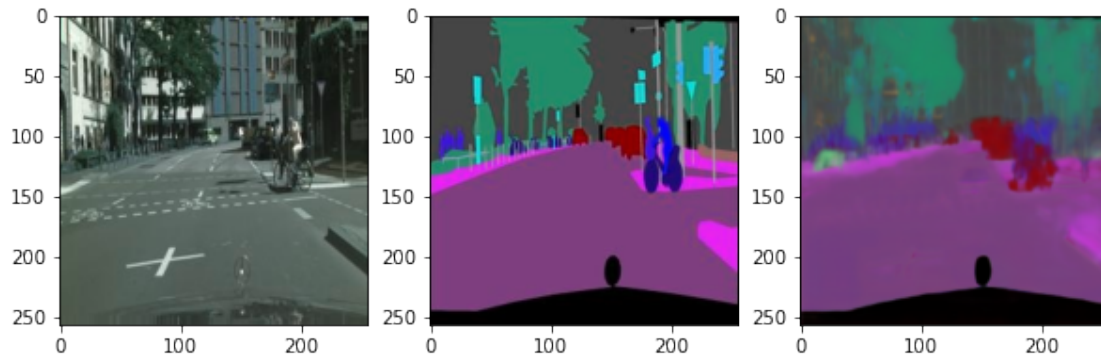
```
[9]: #show the result
pp = model.predict(x_test[:5,:,:,:], batch_size=1)

ni = 5
for k in range(ni):

    plt.figure(figsize=(10,30))
    plt.subplot(ni,3,1+k*3)
    plt.imshow(x_test[k])
    plt.subplot(ni,3,2+k*3)
    plt.imshow(y_test[k])
    plt.subplot(ni,3,3+k*3)
    plt.imshow(pp[k])
```







```
[10]: intersection = np.logical_and(y_test[:5,:,:,:], pp)
union = np.logical_or(y_test[:5,:,:,:], pp)
iou_score = np.sum(intersection) / np.sum(union)
print(iou_score)
```

0.9341532389322916

```
[12]: # Initialize UNET
model2 = UNET(input_shape=(256,256,3), conv_block=conv2d_block, n_filters=32,
↳ dropout=0.5, padding='same', batch_norm=True)
print(model2.summary())
```

Contracting Path
Expanding Path
Model: "model_2"

Layer (type)	Output Shape	Param #	Connected to
=====			
input_3 (InputLayer)	[(None, 256, 256, 3)]	0	

conv2d_50 (Conv2D)	(None, 256, 256, 32)	896	input_3[0][0]

batch_normalization_48 (Batch Normalization)	(None, 256, 256, 32)	128	conv2d_50[0][0]

activation_48 (Activation)	(None, 256, 256, 32)	0	batch_normalization_48[0][0]

conv2d_51 (Conv2D)	(None, 256, 256, 32)	9248	

```

activation_48[0][0]
-----
-----
batch_normalization_49 (BatchNo (None, 256, 256, 32) 128          conv2d_51[0][0]
-----
-----
activation_49 (Activation)      (None, 256, 256, 32) 0
batch_normalization_49[0][0]
-----
-----
max_pooling2d_8 (MaxPooling2D) (None, 128, 128, 32) 0
activation_49[0][0]
-----
-----
dropout_22 (Dropout)           (None, 128, 128, 32) 0
max_pooling2d_8[0][0]
-----
-----
conv2d_52 (Conv2D)              (None, 128, 128, 64) 18496
dropout_22[0][0]
-----
-----
batch_normalization_50 (BatchNo (None, 128, 128, 64) 256          conv2d_52[0][0]
-----
-----
activation_50 (Activation)      (None, 128, 128, 64) 0
batch_normalization_50[0][0]
-----
-----
conv2d_53 (Conv2D)              (None, 128, 128, 64) 36928
activation_50[0][0]
-----
-----
batch_normalization_51 (BatchNo (None, 128, 128, 64) 256          conv2d_53[0][0]
-----
-----
activation_51 (Activation)      (None, 128, 128, 64) 0
batch_normalization_51[0][0]
-----
-----
max_pooling2d_9 (MaxPooling2D) (None, 64, 64, 64)  0
activation_51[0][0]
-----
-----
dropout_23 (Dropout)           (None, 64, 64, 64)  0
max_pooling2d_9[0][0]
-----
-----

```



```

conv2d_54 (Conv2D)                (None, 64, 64, 128) 73856
dropout_23[0][0]

-----

batch_normalization_52 (BatchNo (None, 64, 64, 128) 512          conv2d_54[0][0]
-----

activation_52 (Activation)         (None, 64, 64, 128) 0
batch_normalization_52[0][0]

-----

conv2d_55 (Conv2D)                (None, 64, 64, 128) 147584
activation_52[0][0]

-----

batch_normalization_53 (BatchNo (None, 64, 64, 128) 512          conv2d_55[0][0]
-----

activation_53 (Activation)         (None, 64, 64, 128) 0
batch_normalization_53[0][0]

-----

max_pooling2d_10 (MaxPooling2D) (None, 32, 32, 128) 0
activation_53[0][0]

-----

dropout_24 (Dropout)              (None, 32, 32, 128) 0
max_pooling2d_10[0][0]

-----

conv2d_56 (Conv2D)                (None, 32, 32, 256) 295168
dropout_24[0][0]

-----

batch_normalization_54 (BatchNo (None, 32, 32, 256) 1024          conv2d_56[0][0]
-----

activation_54 (Activation)         (None, 32, 32, 256) 0
batch_normalization_54[0][0]

-----

conv2d_57 (Conv2D)                (None, 32, 32, 256) 590080
activation_54[0][0]

-----

batch_normalization_55 (BatchNo (None, 32, 32, 256) 1024          conv2d_57[0][0]
-----

```

```

activation_55 (Activation)          (None, 32, 32, 256)  0
batch_normalization_55[0][0]
-----

max_pooling2d_11 (MaxPooling2D) (None, 16, 16, 256)  0
activation_55[0][0]
-----

dropout_25 (Dropout)              (None, 16, 16, 256)  0
max_pooling2d_11[0][0]
-----

conv2d_58 (Conv2D)                (None, 16, 16, 512)  1180160
dropout_25[0][0]
-----

batch_normalization_56 (BatchNo (None, 16, 16, 512)  2048      conv2d_58[0][0]
-----

activation_56 (Activation)        (None, 16, 16, 512)  0
batch_normalization_56[0][0]
-----

conv2d_59 (Conv2D)                (None, 16, 16, 512)  2359808
activation_56[0][0]
-----

batch_normalization_57 (BatchNo (None, 16, 16, 512)  2048      conv2d_59[0][0]
-----

activation_57 (Activation)        (None, 16, 16, 512)  0
batch_normalization_57[0][0]
-----

conv2d_transpose_14 (Conv2DTran (None, 32, 32, 256)  1179904
activation_57[0][0]
-----

concatenate_14 (Concatenate)      (None, 32, 32, 512)  0
conv2d_transpose_14[0][0]
activation_55[0][0]
-----

dropout_26 (Dropout)              (None, 32, 32, 512)  0
concatenate_14[0][0]
-----

conv2d_60 (Conv2D)                (None, 32, 32, 256)  1179904

```

dropout_26[0][0]

batch_normalization_58 (BatchNo (None, 32, 32, 256) 1024 conv2d_60[0][0]

activation_58 (Activation) (None, 32, 32, 256) 0
batch_normalization_58[0][0]

conv2d_61 (Conv2D) (None, 32, 32, 256) 590080
activation_58[0][0]

batch_normalization_59 (BatchNo (None, 32, 32, 256) 1024 conv2d_61[0][0]

activation_59 (Activation) (None, 32, 32, 256) 0
batch_normalization_59[0][0]

conv2d_transpose_15 (Conv2DTran (None, 64, 64, 128) 295040
activation_59[0][0]

concatenate_15 (Concatenate) (None, 64, 64, 256) 0
conv2d_transpose_15[0][0]
activation_53[0][0]

dropout_27 (Dropout) (None, 64, 64, 256) 0
concatenate_15[0][0]

conv2d_62 (Conv2D) (None, 64, 64, 128) 295040
dropout_27[0][0]

batch_normalization_60 (BatchNo (None, 64, 64, 128) 512 conv2d_62[0][0]

activation_60 (Activation) (None, 64, 64, 128) 0
batch_normalization_60[0][0]

conv2d_63 (Conv2D) (None, 64, 64, 128) 147584
activation_60[0][0]

```

-----
batch_normalization_61 (BatchNo (None, 64, 64, 128) 512          conv2d_63[0] [0]
-----

-----
activation_61 (Activation)      (None, 64, 64, 128) 0
batch_normalization_61[0] [0]
-----

-----
conv2d_transpose_16 (Conv2DTran (None, 128, 128, 64) 73792
activation_61[0] [0]
-----

-----
concatenate_16 (Concatenate)    (None, 128, 128, 128 0
conv2d_transpose_16[0] [0]
activation_51[0] [0]
-----

-----
dropout_28 (Dropout)           (None, 128, 128, 128 0
concatenate_16[0] [0]
-----

-----
conv2d_64 (Conv2D)             (None, 128, 128, 64) 73792
dropout_28[0] [0]
-----

-----
batch_normalization_62 (BatchNo (None, 128, 128, 64) 256          conv2d_64[0] [0]
-----

-----
activation_62 (Activation)      (None, 128, 128, 64) 0
batch_normalization_62[0] [0]
-----

-----
conv2d_65 (Conv2D)             (None, 128, 128, 64) 36928
activation_62[0] [0]
-----

-----
batch_normalization_63 (BatchNo (None, 128, 128, 64) 256          conv2d_65[0] [0]
-----

-----
activation_63 (Activation)      (None, 128, 128, 64) 0
batch_normalization_63[0] [0]
-----

-----
conv2d_transpose_17 (Conv2DTran (None, 256, 256, 32) 18464
activation_63[0] [0]
-----

-----
concatenate_17 (Concatenate)    (None, 256, 256, 64) 0

```

```

conv2d_transpose_17[0][0]
activation_49[0][0]

-----

dropout_29 (Dropout)                (None, 256, 256, 64) 0
concatenate_17[0][0]

-----

conv2d_66 (Conv2D)                  (None, 256, 256, 32) 18464
dropout_29[0][0]

-----

batch_normalization_64 (BatchNo (None, 256, 256, 32) 128      conv2d_66[0][0]
-----

activation_64 (Activation)           (None, 256, 256, 32) 0
batch_normalization_64[0][0]

-----

conv2d_67 (Conv2D)                  (None, 256, 256, 32) 9248
activation_64[0][0]

-----

batch_normalization_65 (BatchNo (None, 256, 256, 32) 128      conv2d_67[0][0]
-----

activation_65 (Activation)           (None, 256, 256, 32) 0
batch_normalization_65[0][0]

-----

conv2d_68 (Conv2D)                  (None, 256, 256, 3) 99
activation_65[0][0]
=====
=====
Total params: 8,642,339
Trainable params: 8,636,451
Non-trainable params: 5,888

-----

None

```

```
[13]: # Compile model with specified optimizer and loss
model2.compile(optimizer='adam', loss='mse')
```

```
[14]: # Track model history as it trains
import tensorflow as tf
print(tf.__version__)
```

```
h2 = model2.fit(x_train, y_train, epochs=5, shuffle=True, batch_size=1,
    ↪ validation_data=(x_val, y_val))
```

2.1.0

Train on 2677 samples, validate on 298 samples

Epoch 1/5

2677/2677 [=====] - 81s 30ms/sample - loss: 0.0237 -
val_loss: 0.0228

Epoch 2/5

2677/2677 [=====] - 78s 29ms/sample - loss: 0.0180 -
val_loss: 0.0213

Epoch 3/5

2677/2677 [=====] - 77s 29ms/sample - loss: 0.0160 -
val_loss: 0.0180

Epoch 4/5

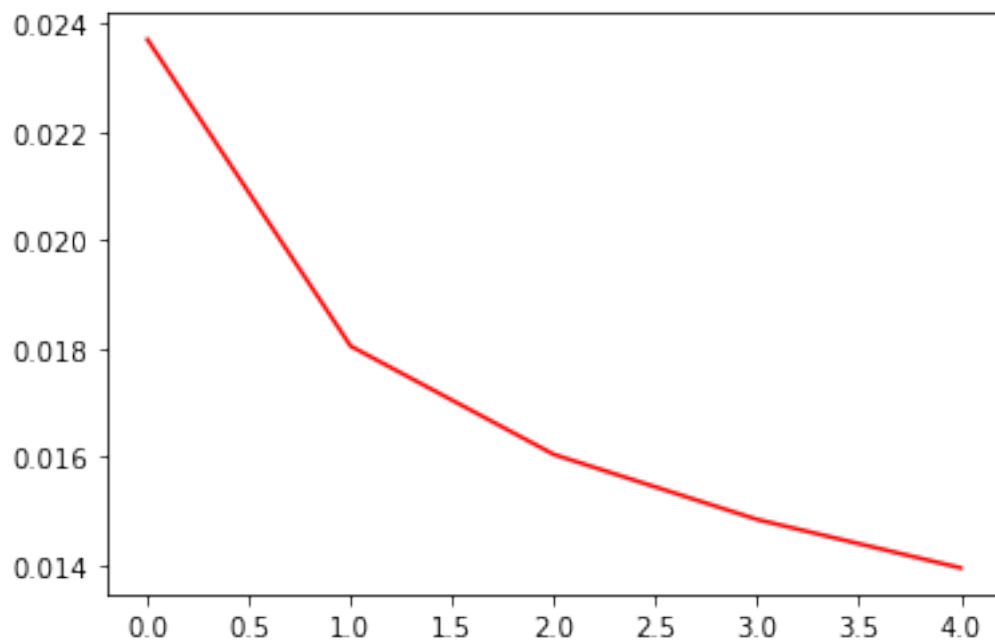
2677/2677 [=====] - 78s 29ms/sample - loss: 0.0148 -
val_loss: 0.0211

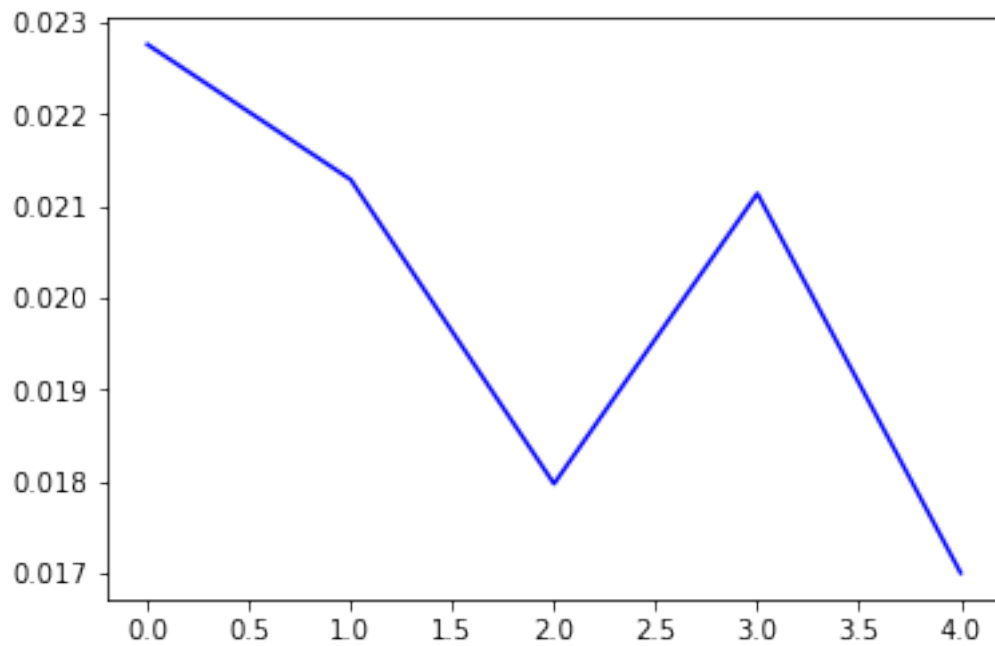
Epoch 5/5

2677/2677 [=====] - 77s 29ms/sample - loss: 0.0140 -
val_loss: 0.0170

[15]: *# Print results for training MSE and validation MSE*

```
plt.plot(h2.history['loss'], 'r')  
plt.show()  
plt.plot(h2.history['val_loss'], 'b')  
plt.show()
```

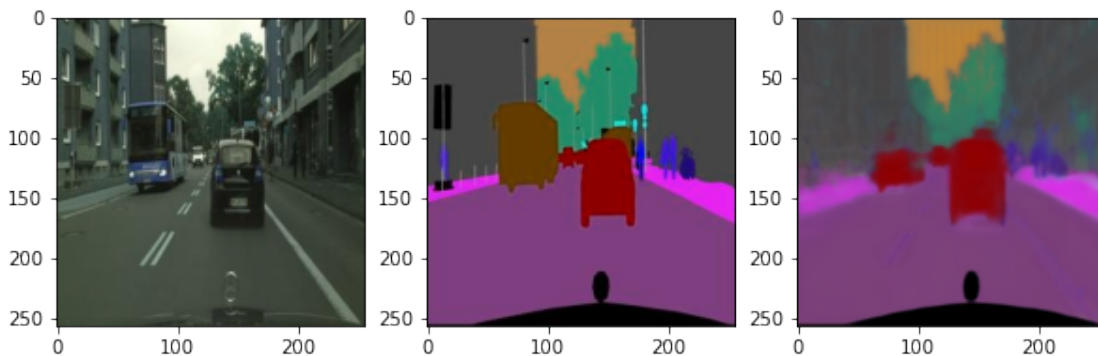


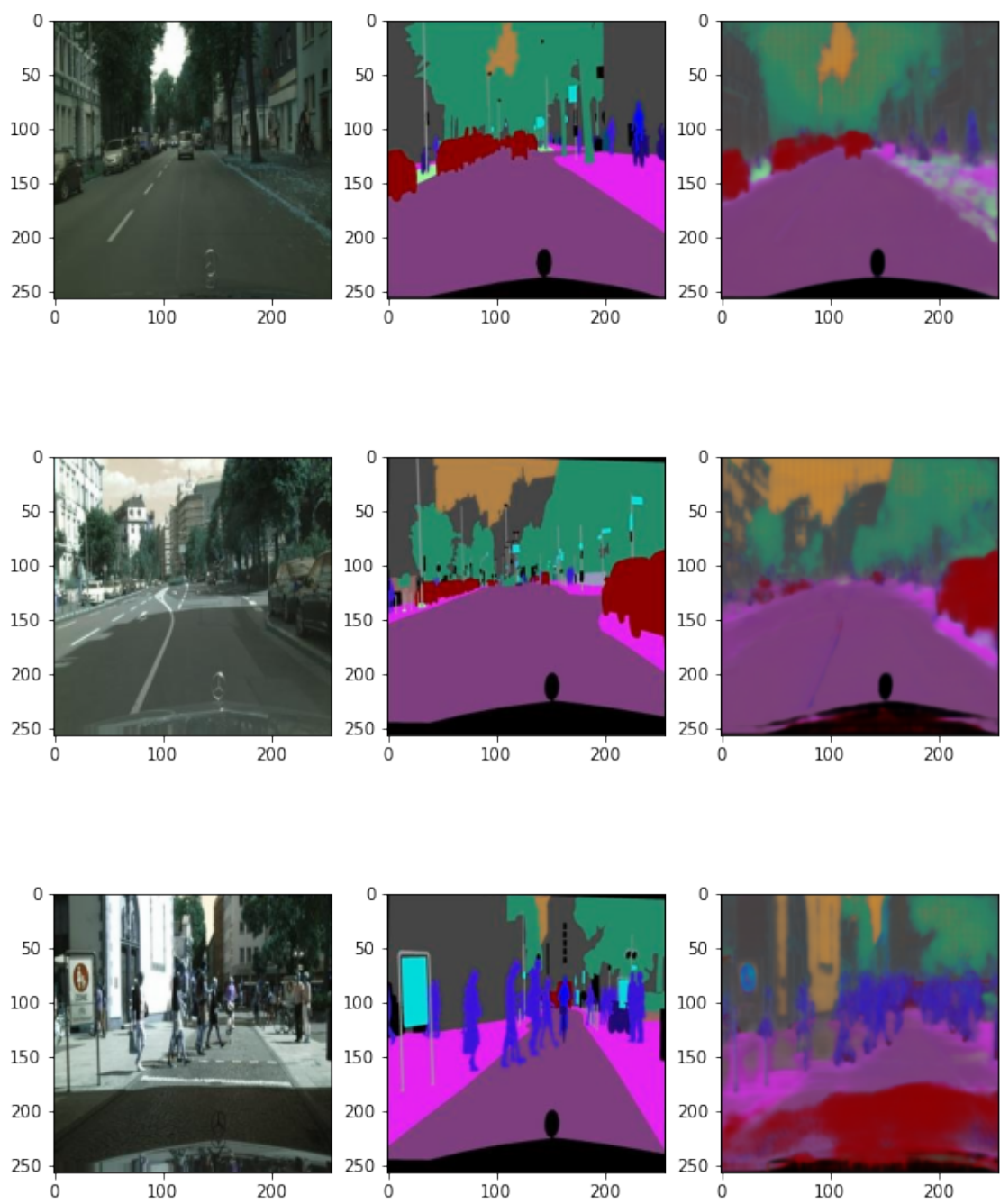


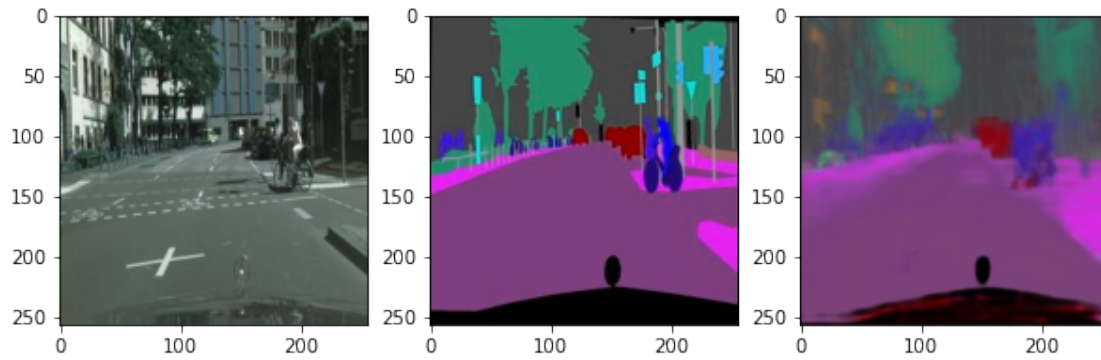
```
[18]: #show the result
pp2 = model2.predict(x_test[:5,:,:,:], batch_size=1)

ni = 5
for k in range(ni):

    plt.figure(figsize=(10,30))
    plt.subplot(ni,3,1+k*3)
    plt.imshow(x_test[k])
    plt.subplot(ni,3,2+k*3)
    plt.imshow(y_test[k])
    plt.subplot(ni,3,3+k*3)
    plt.imshow(pp2[k])
```







```
[17]: intersection = np.logical_and(y_test[:5,:,:,:], pp2)
union = np.logical_or(y_test[:5,:,:,:], pp2)
iou_score = np.sum(intersection) / np.sum(union)
print(iou_score)
```

0.9341532389322916

```
[36]: diff = y_test[1] - pp[1]
m_norm = np.sum(abs(diff))
print(m_norm)

diff2 = y_test[1] - pp2[1]
m_norm2 = np.sum(abs(diff2))
print(m_norm2)
```

9526.393725810676
11908.099578694435