AN2DL – Homework 1

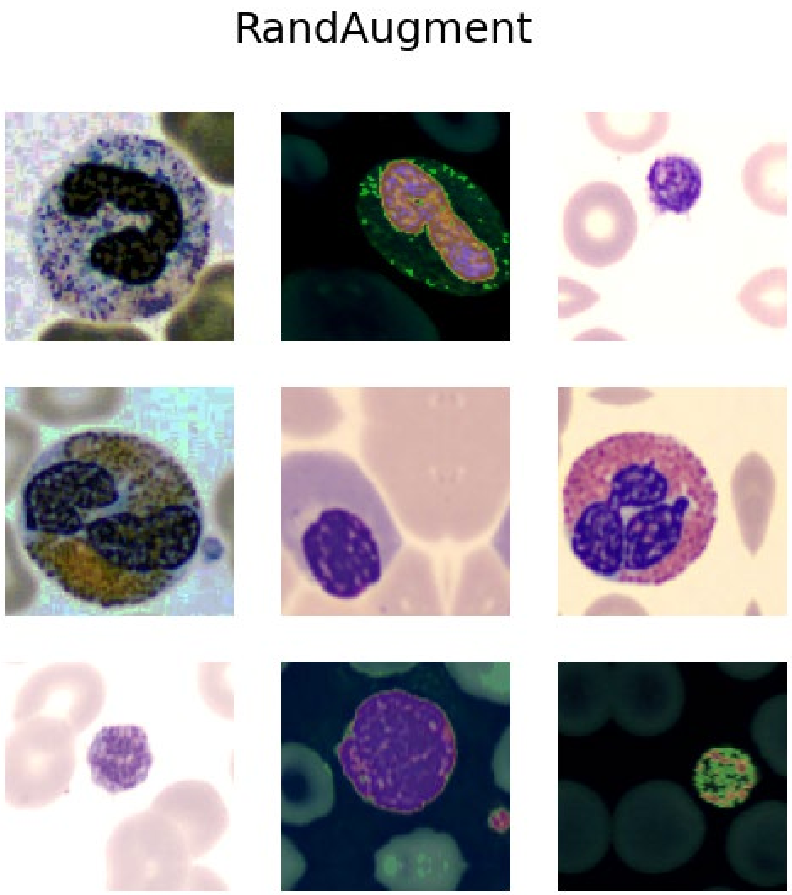
ANNamo bene

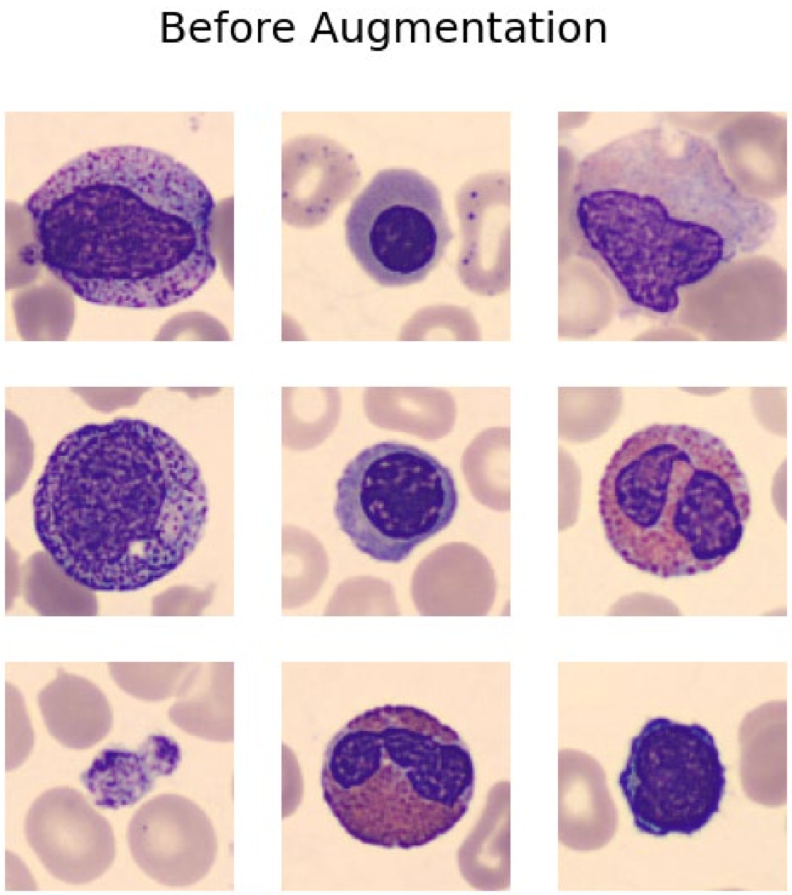
Michelangelo Stasi, Nicolò Tocalli, Elia Pontiggia

Key idea 1: Image Augmentation

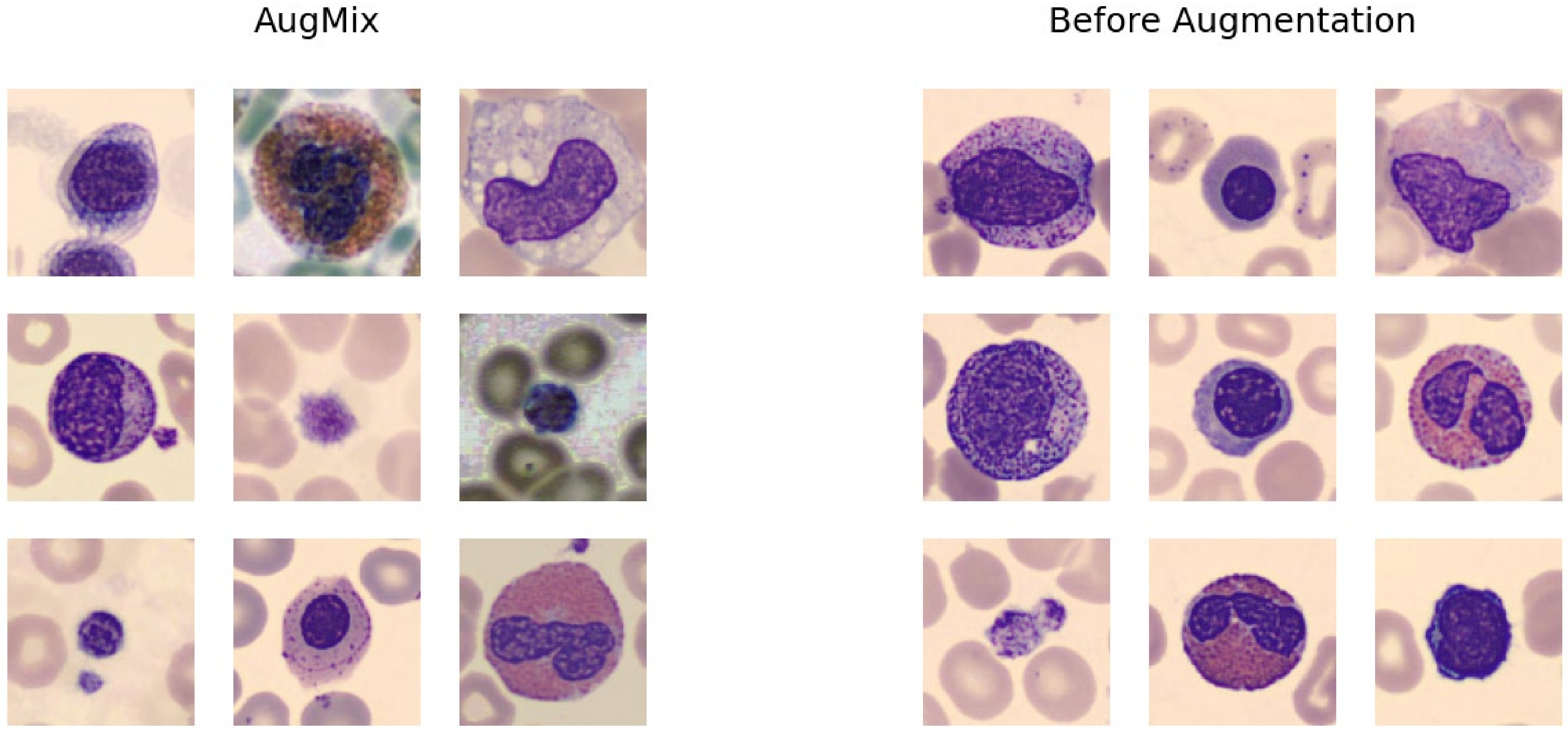
Corporate needs you to find the difference between this picture and this picture
Our model:
They're the same picture

keras_cv.layers.RandAugment()





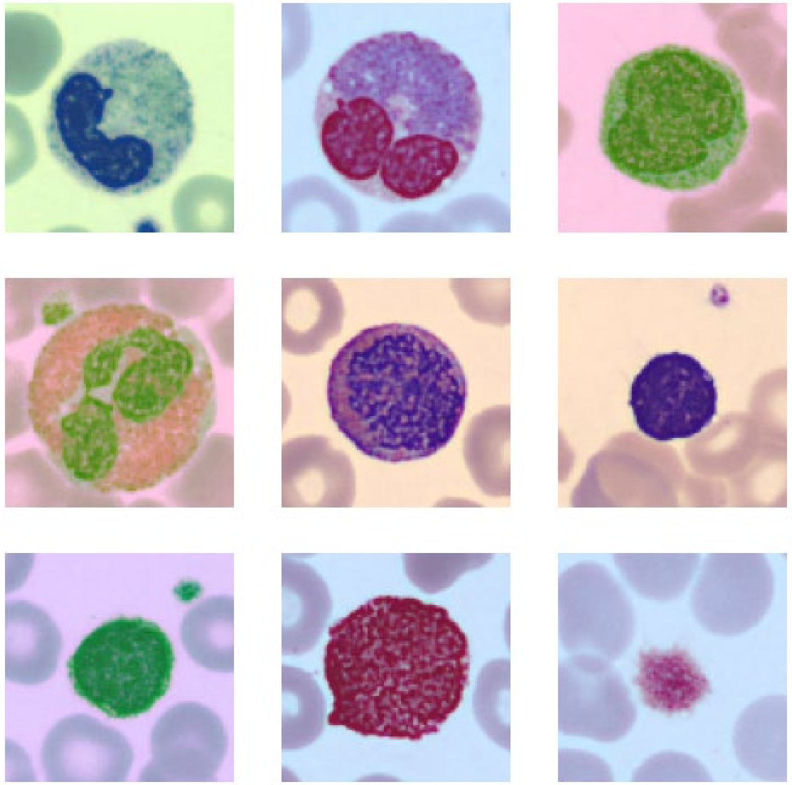
keras_cv.layers.AugMix()

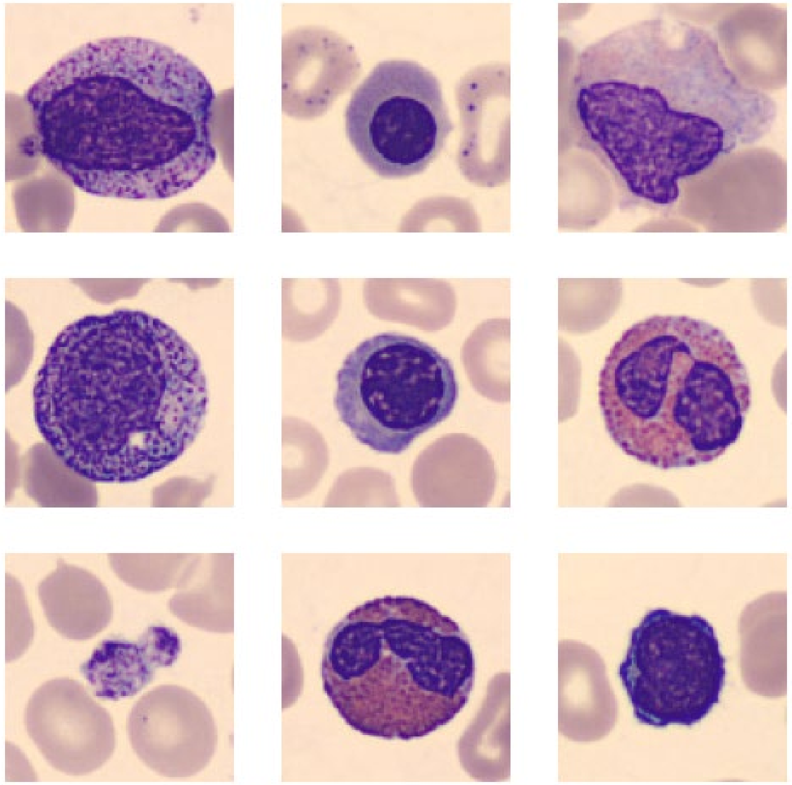


AugMix

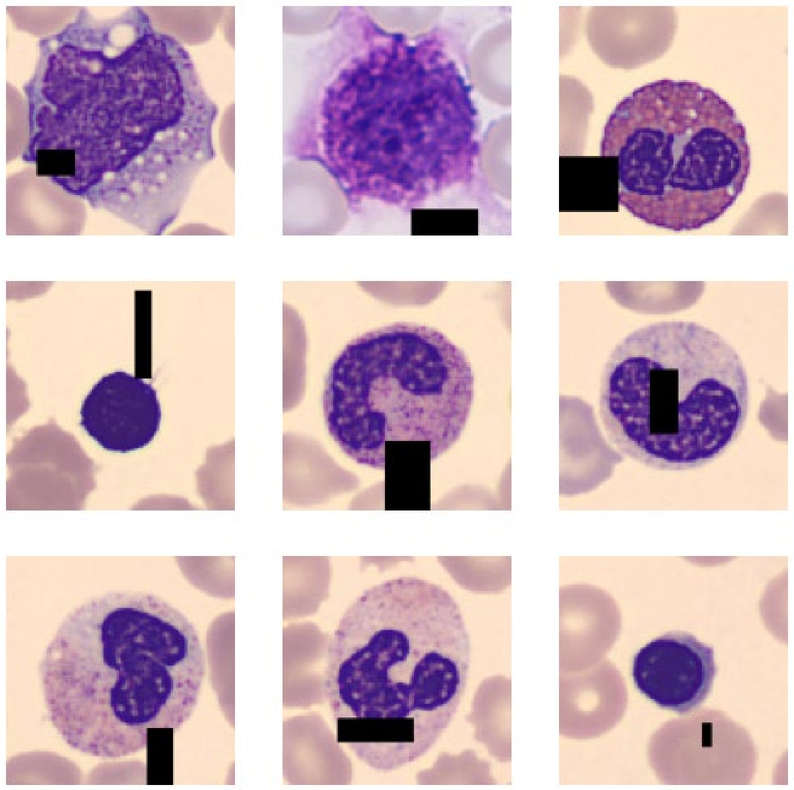
Before Augmentation

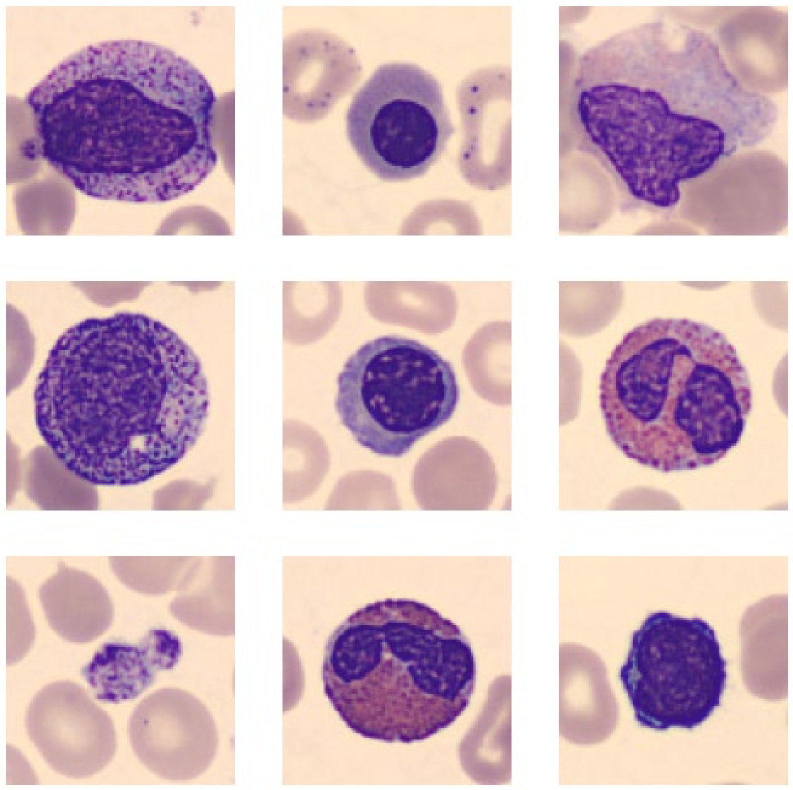
keras_cv.layers.ChannelShuffle()


After Channel Shuffle

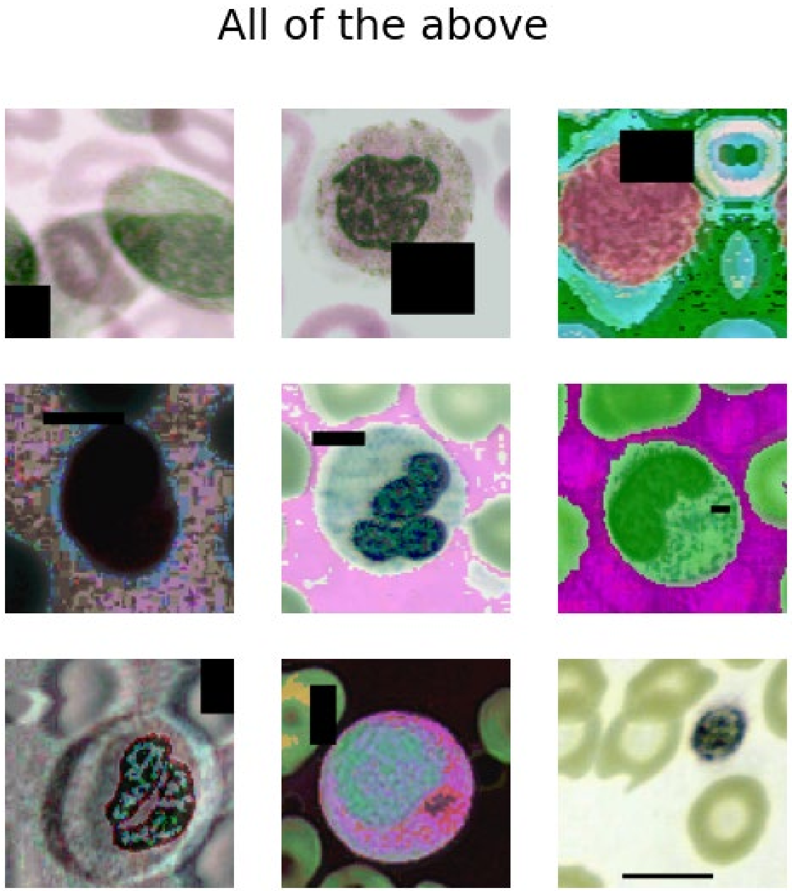

Before Augmentation

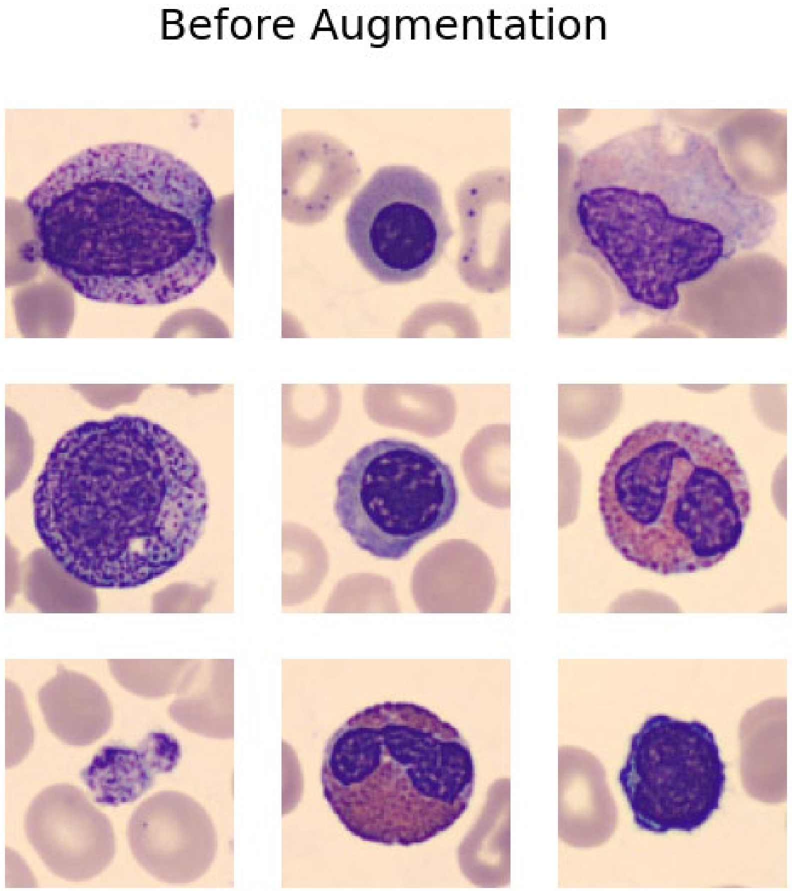
keras_cv.layers.RandomCutout()


After Cutout

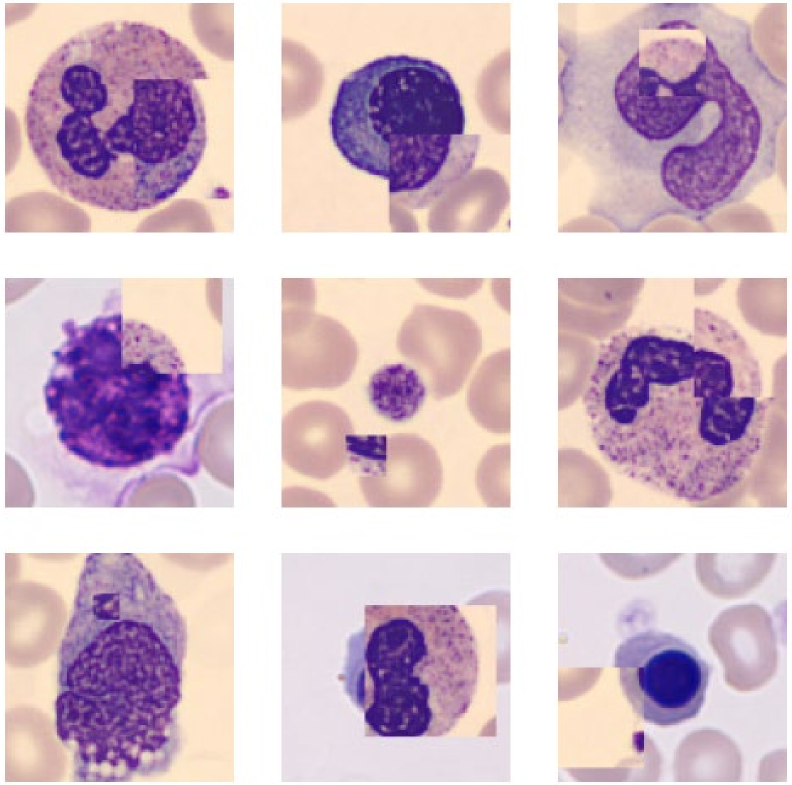

Before Augmentation

All of the above (same label)

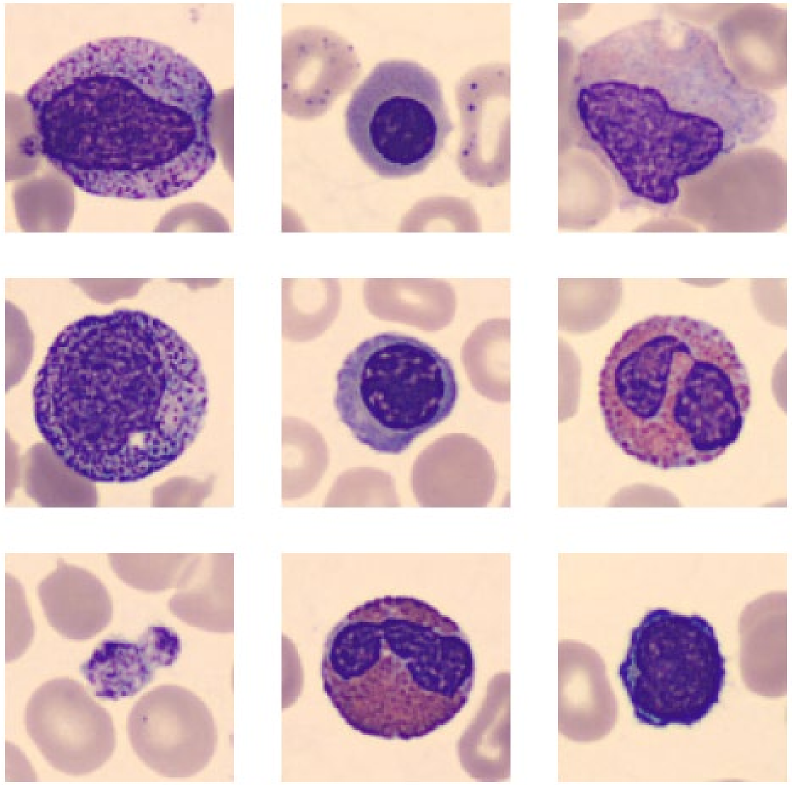




keras_cv.layers.CutMix(),
keras_cv.layers.MixUp()



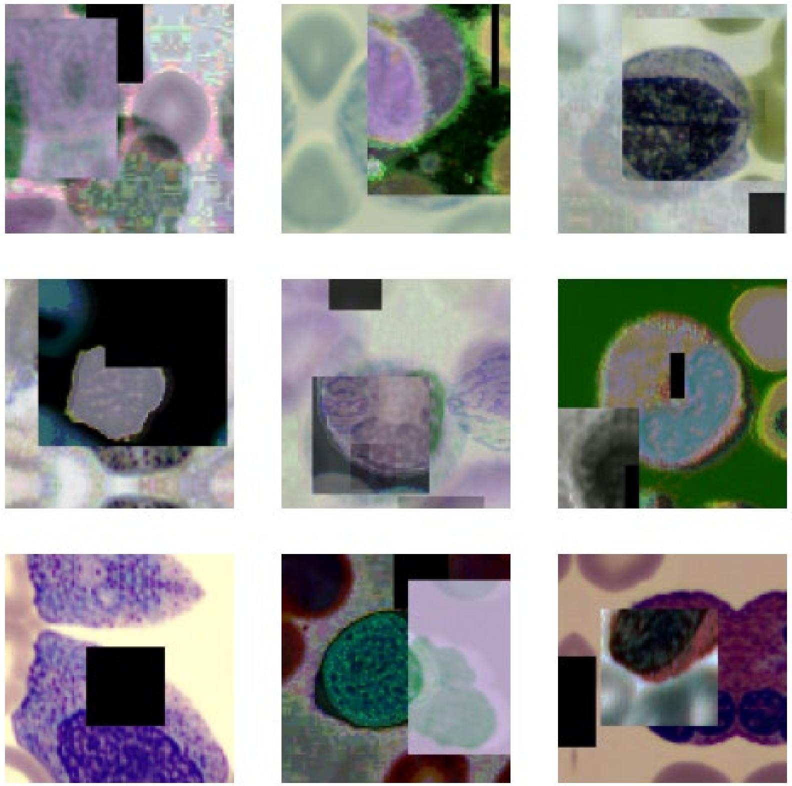
After CutMix and MixUp



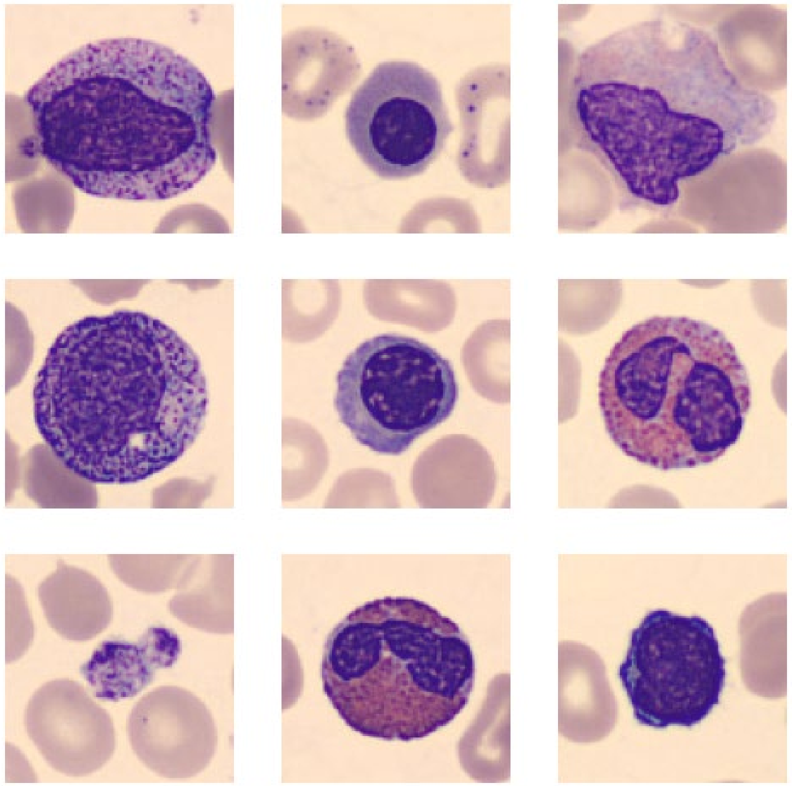
Before Augmentation

Everything put togheter

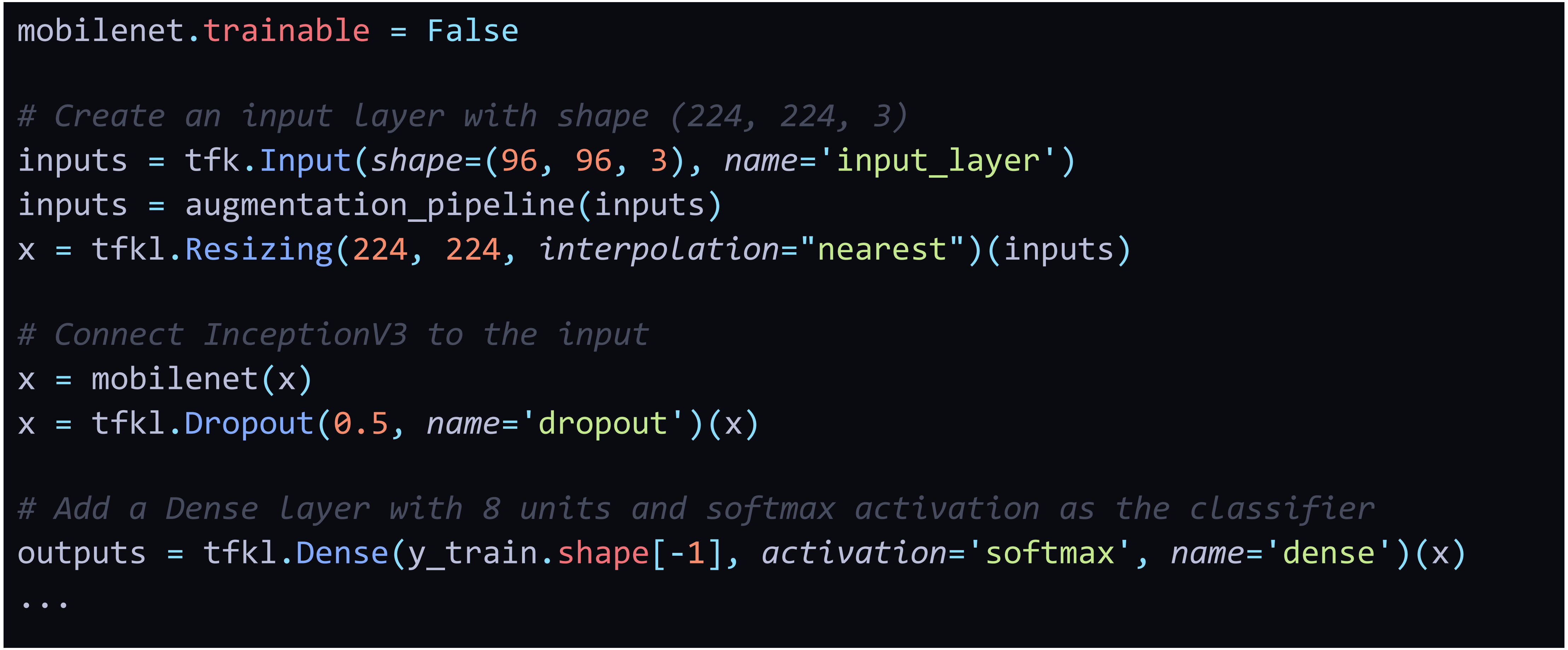
All



Before Augmentation

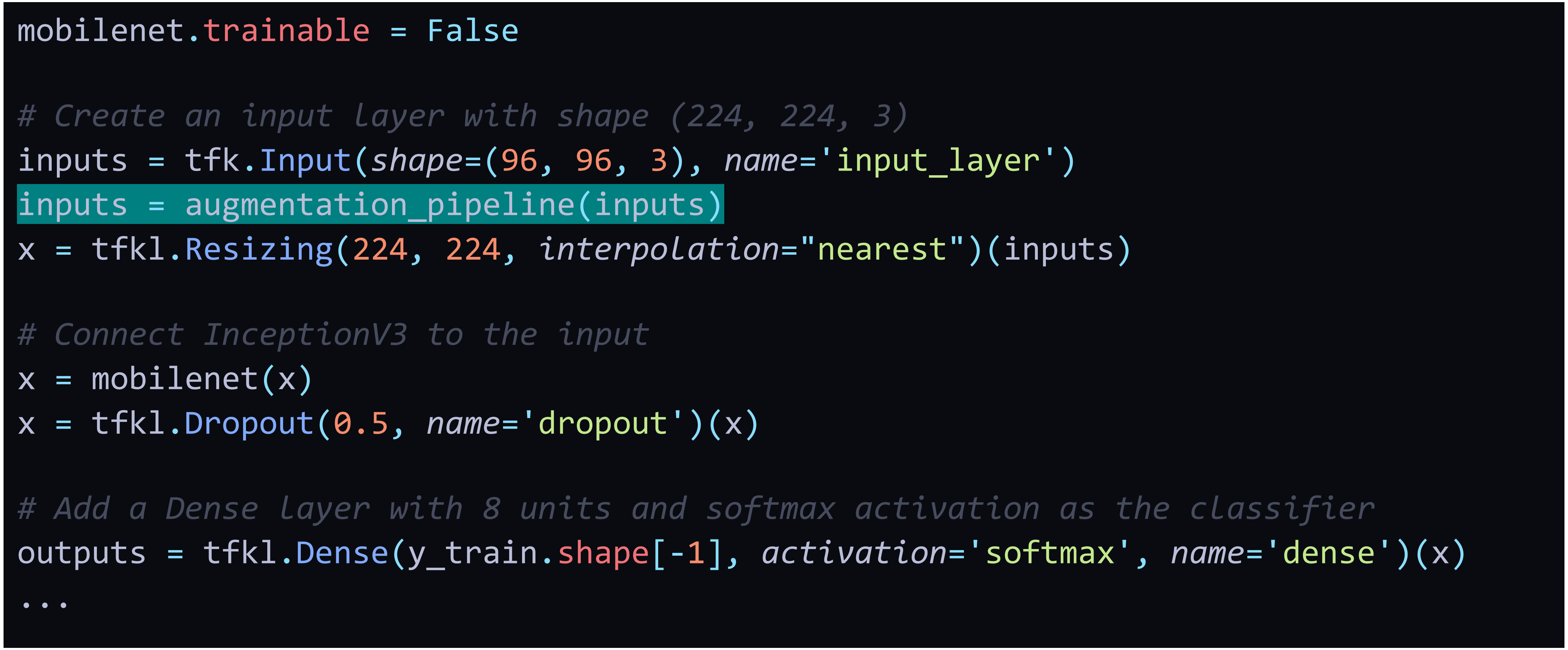


Both «offline» and «online»



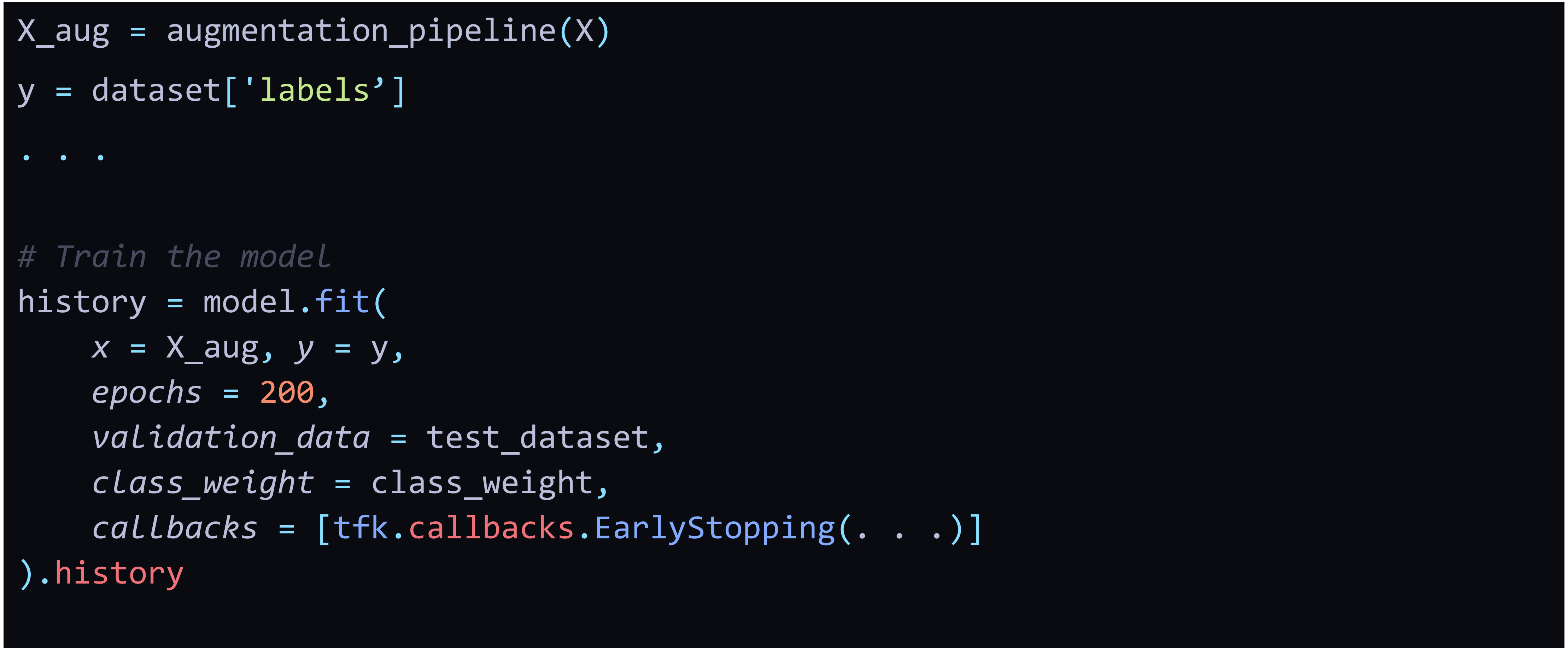
```
mobilenet.trainable = False

# Create an input layer with shape (224, 224, 3)
inputs = tfk.Input(shape=(96, 96, 3), name='input_layer')
inputs = augmentation_pipeline(inputs)
x = tfkl.Resizing(224, 224, interpolation="nearest")(inputs)

# Connect InceptionV3 to the input
x = mobilenet(x)
x = tfkl.Dropout(0.5, name='dropout')(x)

# Add a Dense layer with 8 units and softmax activation as the classifier
outputs = tfkl.Dense(y_train.shape[-1], activation='softmax', name='dense')(x)
...
```

Both «offline» and «online»



```
mobilenet.trainable = False

# Create an input layer with shape (224, 224, 3)
inputs = tfk.Input(shape=(96, 96, 3), name='input_layer')
inputs = augmentation_pipeline(inputs)
x = tfkl.Resizing(224, 224, interpolation="nearest")(inputs)

# Connect InceptionV3 to the input
x = mobilenet(x)
x = tfkl.Dropout(0.5, name='dropout')(x)

# Add a Dense layer with 8 units and softmax activation as the classifier
outputs = tfkl.Dense(y_train.shape[-1], activation='softmax', name='dense')(x)
...
```

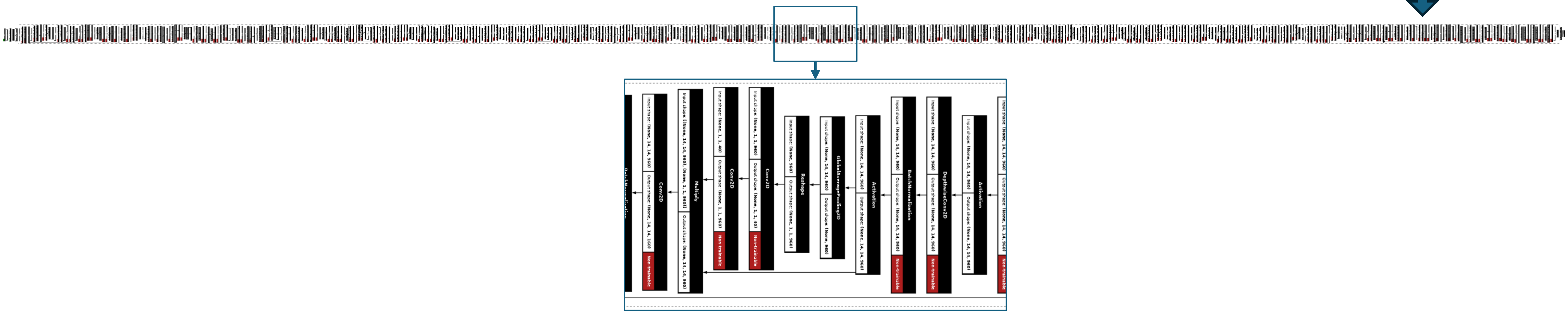
Both «offline» and «online»



```
X_aug = augmentation_pipeline(X)

y = dataset['labels’]

. . .


# Train the model
history = model.fit(
    x = X_aug, y = y,
    epochs = 200,
    validation_data = test_dataset,
    class_weight = class_weight,
    callbacks = [tfk.callbacks.EarlyStopping(. . .)]
).history
```

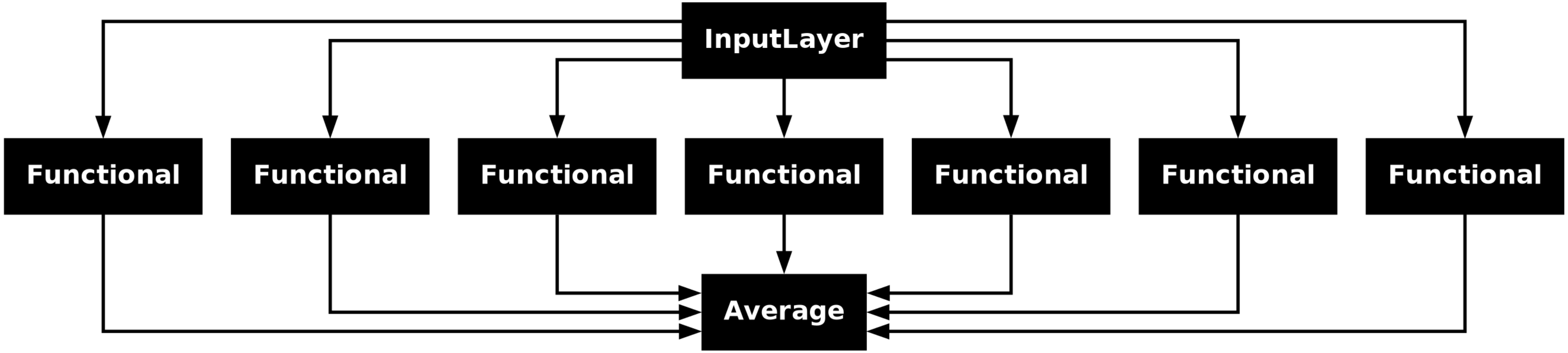
Key Idea 2: Transfer Learning

- EfficientNet:
  - V2S (88MB; 83,9% top-1; 96,7% top-5; 21,6M params)
  - V2M
  - V2L
  - B2 (discarded)
- Vgg16 (discarded)



Input shape: (None, 14, 14, 960)
Conv2D
Output shape: (None, 14, 14, 160)
Non-trainable
Input shape: [(None, 14, 14, 960), (None, 1, 1, 960)]
Multiply
Output shape: (None, 14, 14, 960)
Input shape: (None, 1, 1, 40)
Conv2D
Output shape: (None, 1, 1, 960)
Non-trainable
Input shape: (None, 1, 1, 960)
Conv2D
Output shape: (None, 1, 1, 40)
Non-trainable
Input shape: (None, 960)
Reshape
Output shape: (None, 1, 1, 960)
Input shape: (None, 14, 14, 960)
GlobalAveragePooling2D
Output shape: (None, 960)
Input shape: (None, 14, 14, 960)
Activation
Output shape: (None, 14, 14, 960)
Input shape: (None, 14, 14, 960)
BatchNormalization
Output shape: (None, 14, 14, 960)
Non-trainable
Input shape: (None, 14, 14, 960)
DepthwiseConv2D
Output shape: (None, 14, 14, 960)
Non-trainable
Input shape: (None, 14, 14, 960)
Activation
Output shape: (None, 14, 14, 960)
Input shape: (None, 14, 14, 960)
Output shape: (None, 14, 14, 960)
Non-trainable

Key Idea 3: Ensemble


InputLayer
Functional
Functional
Functional
Functional
Functional
Functional
Functional
Average