Stan Rares-Gabriel, 431B

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(460, 700, 3)),
    MaxPooling2D((2, 2)),
    BatchNormalization(),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    BatchNormalization(),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    BatchNormalization(),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(2, activation='softmax')
])
	Epoci - 10
	Batch_size - 32
		accuracy: 0.8220183253288269
		
		loss: 5.213585376739502

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(460, 700, 3)),
    MaxPooling2D((2, 2)),
    BatchNormalization(),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    BatchNormalization(),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    BatchNormalization(),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(2, activation='softmax')
])
	Epoci - 5
	Batch_size - 32
		accuracy: 0.631192684173584
		
		loss: 11.369023323059082

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(460, 700, 3)),
    MaxPooling2D((2, 2)),
    BatchNormalization(),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    BatchNormalization(),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    BatchNormalization(),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(2, activation='softmax')
])
	Epoci - 10
	Batch_size - 32
		accuracy: 0.8165137767791748
		
		loss: 17.243104934692383

Stan Rares-Gabriel, 431B

`model = Sequential([Conv2D(32, (3, 3), activation='relu', input_shape=(460, 700, 3)), MaxPooling2D((2, 2)), BatchNormalization(), Conv2D(64, (3, 3), activation='relu'), MaxPooling2D((2, 2)), BatchNormalization(), Conv2D(64, (3, 3), activation='relu'), MaxPooling2D((2, 2)), BatchNormalization(), Flatten(), Dense(64, activation='relu'), Dense(2, activation='softmax')])`	Epoci - 5 Batch_size - 32	**accuracy: 0.7412844300270081** **loss: 3.6607916355133057**
`base_model = VGG16(weights='/kaggle/input/vgg16rares/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5', include_top=False, input_shape=(460, 700, 3)) v16_model = Sequential([ base_model, Flatten(), Dense(256, activation='relu'), Dense(2, activation='softmax') ])`	Epoci - 2 Batch_size - 32	**accuracy: 0.8238531947135925** **loss: 0.5997423529624939**
`base_model = VGG16(weights='/kaggle/input/vgg16rares/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5', include_top=False, input_shape=(460, 700, 3)) v16_model = Sequential([ base_model, Flatten(), Dense(256, activation='relu'), Dense(2, activation='softmax') ])`	Epoci - 5 Batch_size - 32	**accuracy: 0.831192672252655** **loss: 0.6366472840309143**

Rezultat final(VGG-16, Epochs=5, batch_size=32):

- **accuracy: 0.831192672252655**
- **loss: 0.6366472840309143**