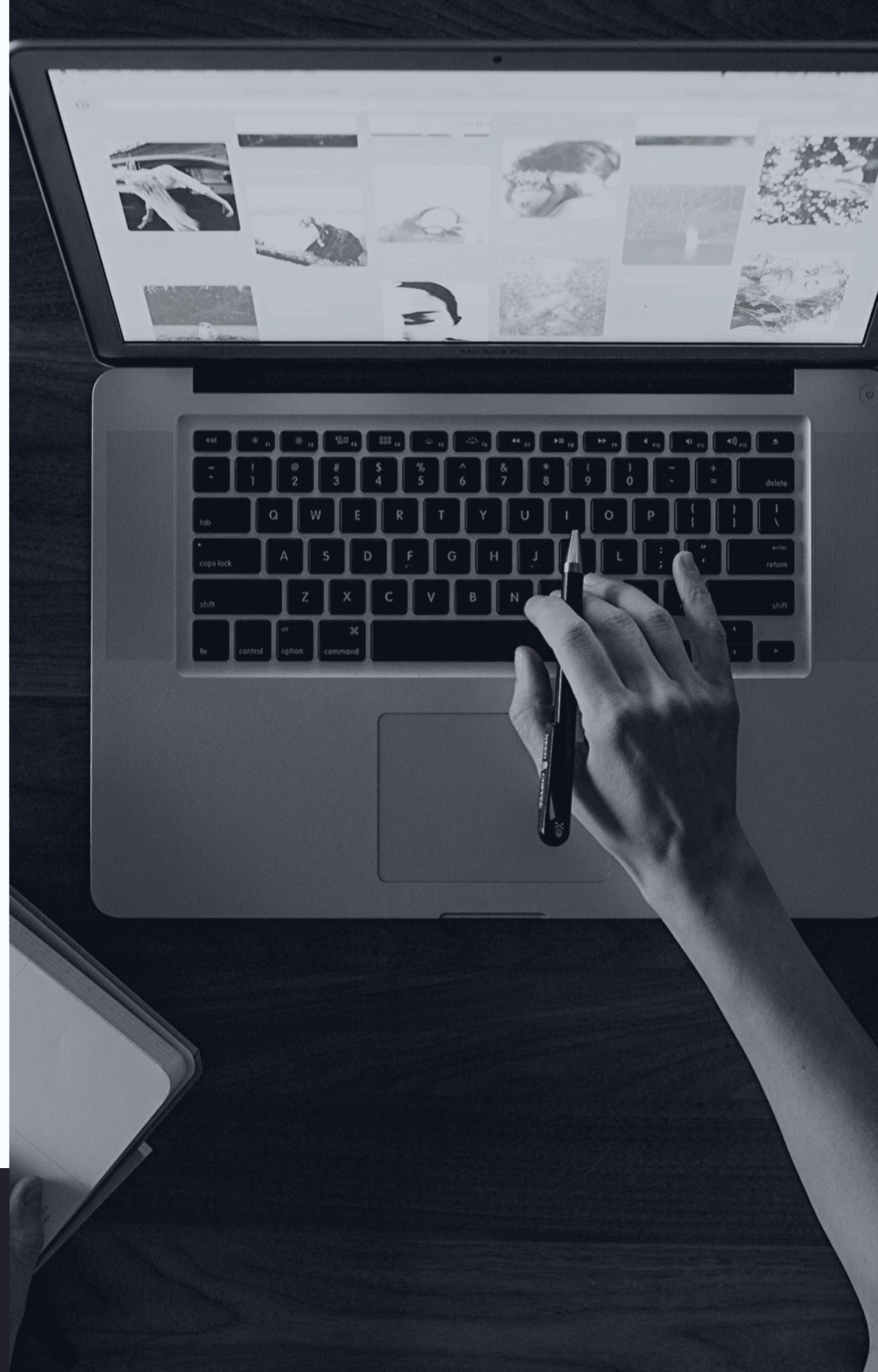


KELompok - 4

ANIMAL CLASSIFICATION

Tugas Besar Machine Learning





ANGGOTA

Muhammad Rizqi Mubarak
Yusril Firza
Muh Rico Andreas
Abdul Manan
Muhammad Wahyu Putra Indi
Yusuf Azam Sya'bani



CNN

CONVOLUTIONAL NEURAL NETWORK



CNN

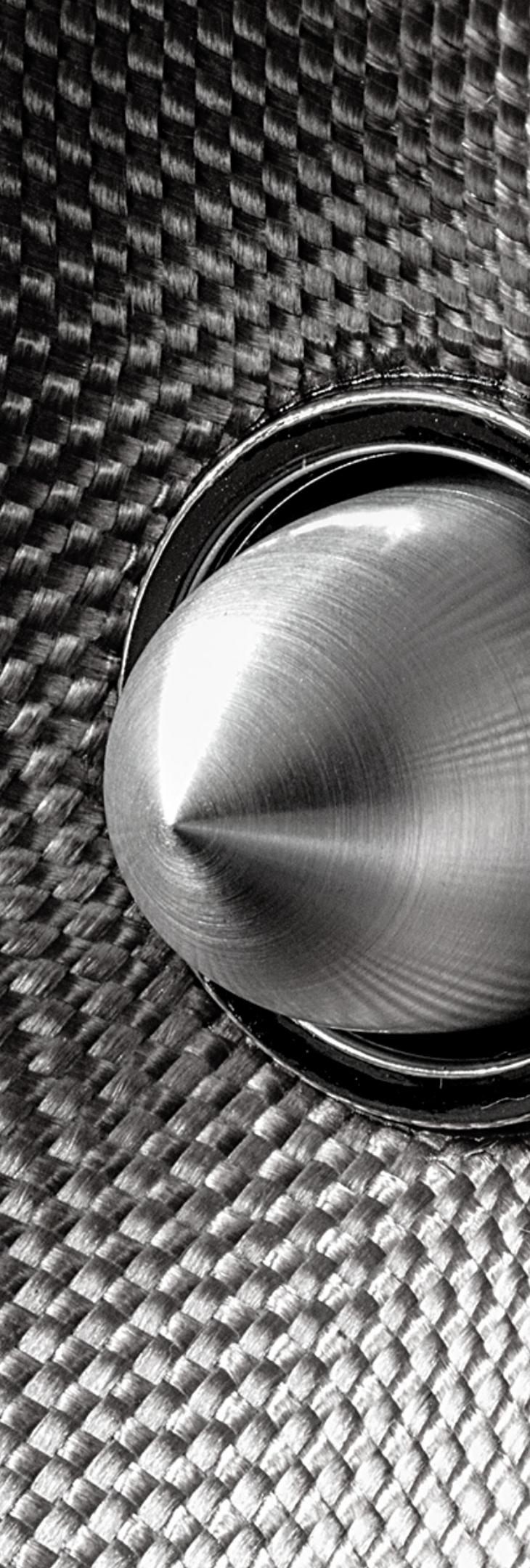
Convolutional Neural Network adalah salah satu metode machine learning dari pengembangan Multi Layer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi.

CNN

CNN termasuk dalam jenis Deep Neural Network karena dalamnya tingkat jaringan dan banyak diimplementasikan dalam data citra.

Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi

CNN merupakan pengembangan lebih lanjut dari MLP karena menggunakan metode yang mirip dengan dimensi yang lebih banyak.



CNN LAYER



Input Layer



Hidden Layer



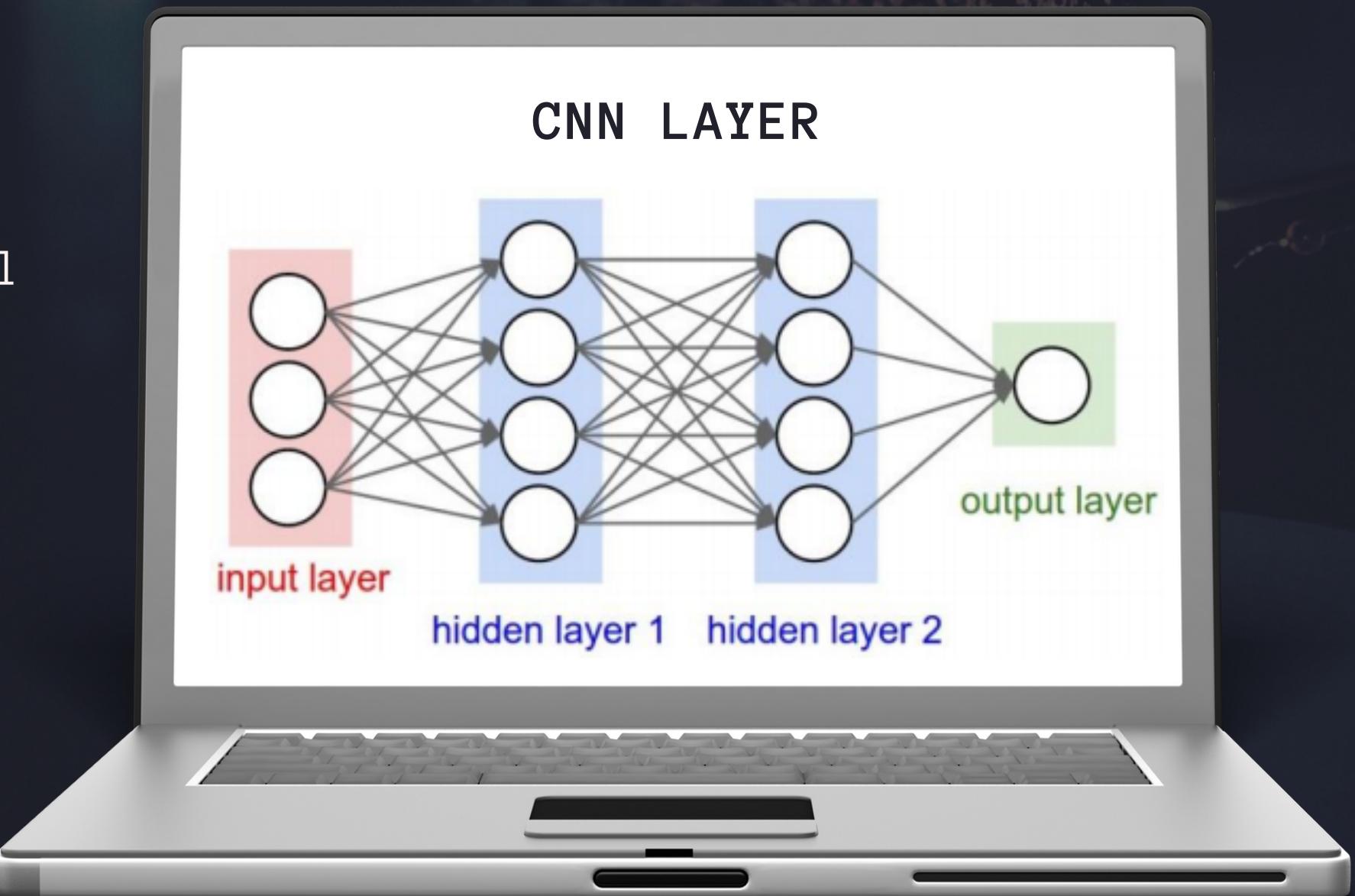
Output Layer

Sebuah MLP seperti pada disamping memiliki i layer (kotak merah dan biru) dengan masing-masing layer berisi ji neuron (lingkaran putih). MLP menerima input data satu dimensi dan mempropagasi data tersebut pada jaringan hingga menghasilkan output.

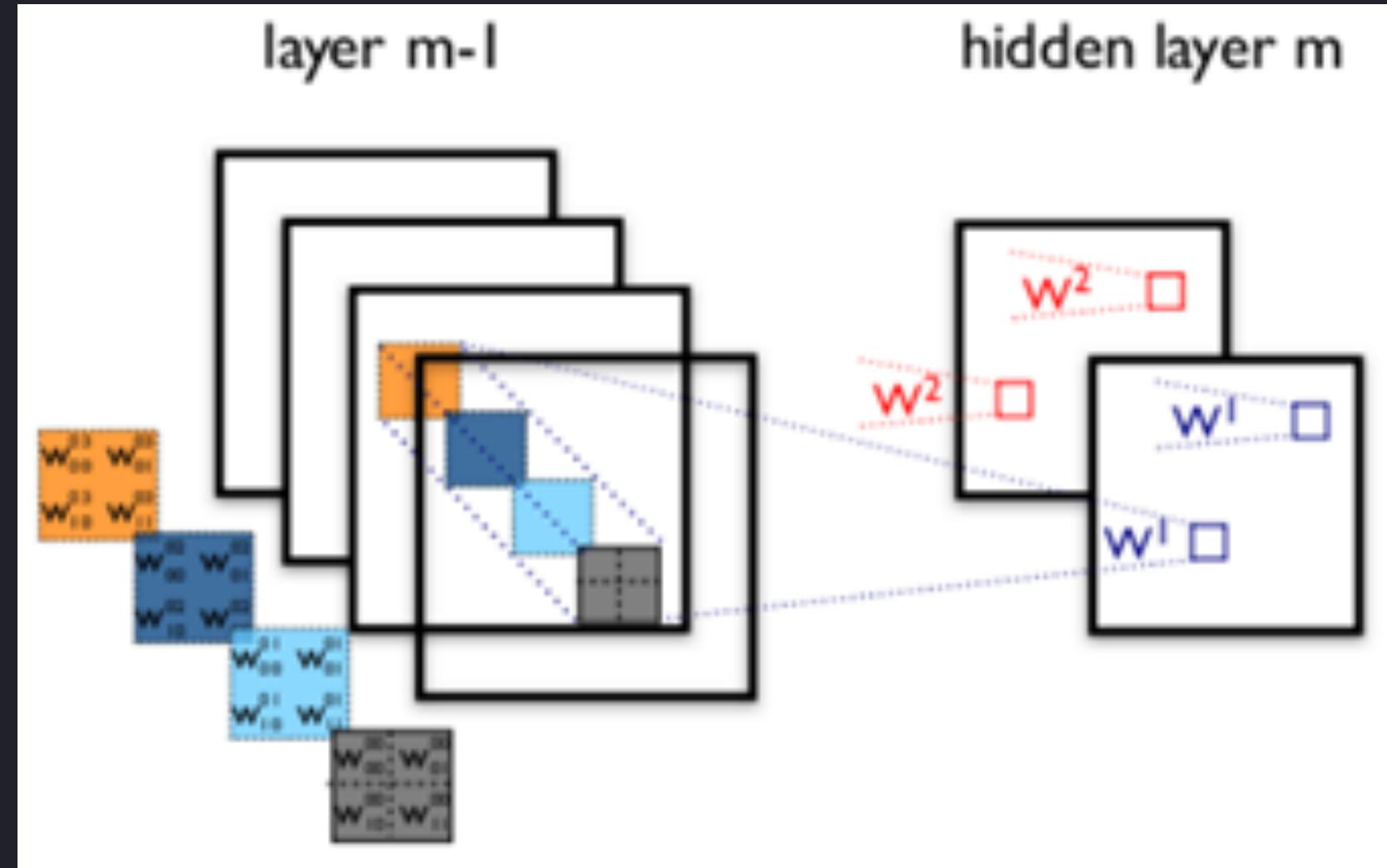
Setiap hubungan antar neuron pada dua layer yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode.

Disetiap data input pada layer dilakukan operasi linear dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi non linear yang disebut sebagai fungsi aktivasi.

Data yang dipropagasi pada CNN adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda.



CONT.



Operasi linear pada CNN menggunakan operasi konvolusi, dengan bobot yang tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi seperti pada gambar di atas.



Hasil Pengerjaan

Tugas Besar Machine Learning

Source code

Kaggle

<https://kaggle.com/min4tozaki/animal-classification>

Google Colab

<https://min4tozaki.me/tubes-ml>

Github

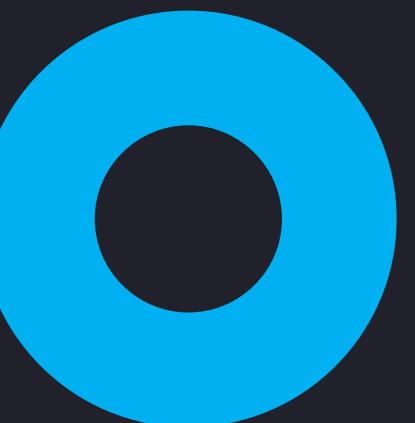
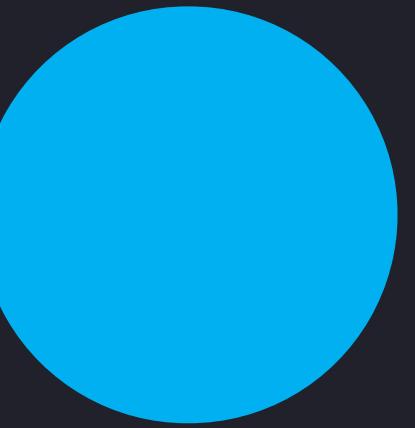
https://github.com/min4tozaki/tubes_ml



— Dataset —

Kaggle

<https://www.kaggle.com/alessiocorrado99/animals10>



TOOLS

PYTHON



JUPYTER
NOTEBOOK



GOOGLE
COLAB



KAGGLE
KERNEL



Python Requirements

TENSORFLOW



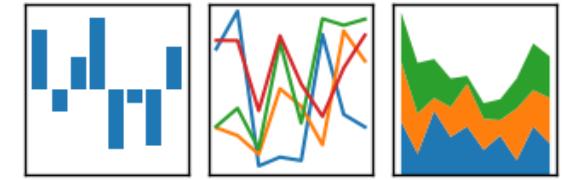
KERAS



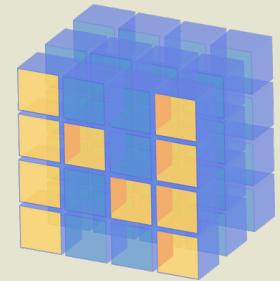
PANDAS

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



NUMPY



OPENCV



SCIKIT-LEARN



Tensorflow Models

```
base_model = applications.VGG16(  
    weights='imagenet',  
    include_top=False,  
    input_shape=(img_rows, img_cols, img_channel))  
  
add_model = Sequential()  
add_model.add(Flatten(input_shape=base_model.output_shape[1:]))  
add_model.add(Dense(256, activation='relu'))  
add_model.add(Dense(10, activation='softmax'))  
  
model = Model(  
    inputs=base_model.input,  
    outputs=add_model(base_model.output))  
  
model.compile(  
    loss='binary_crossentropy',  
    optimizer=optimizers.SGD(lr=1e-4, momentum=0.9),  
    metrics=['accuracy'])  
  
batch_size = 32  
epochs = 50
```

TENSORFLOW

For this model is highly recommended using GPU because the process is much faster then using CPU for process

GPU VS CPU

GPU

40s - 60s per epoch

CPU

20m - 30m per epoch



OUTPUT MODELS

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
sequential_1 (Sequential)	(None, 10)	6425354
Total params: 21,140,042		
Trainable params: 21,140,042		
Non-trainable params: 0		

TRAINING INPUT

data train = 4000 (400/animal)

data test = 1000 (100/animal)

Training Process



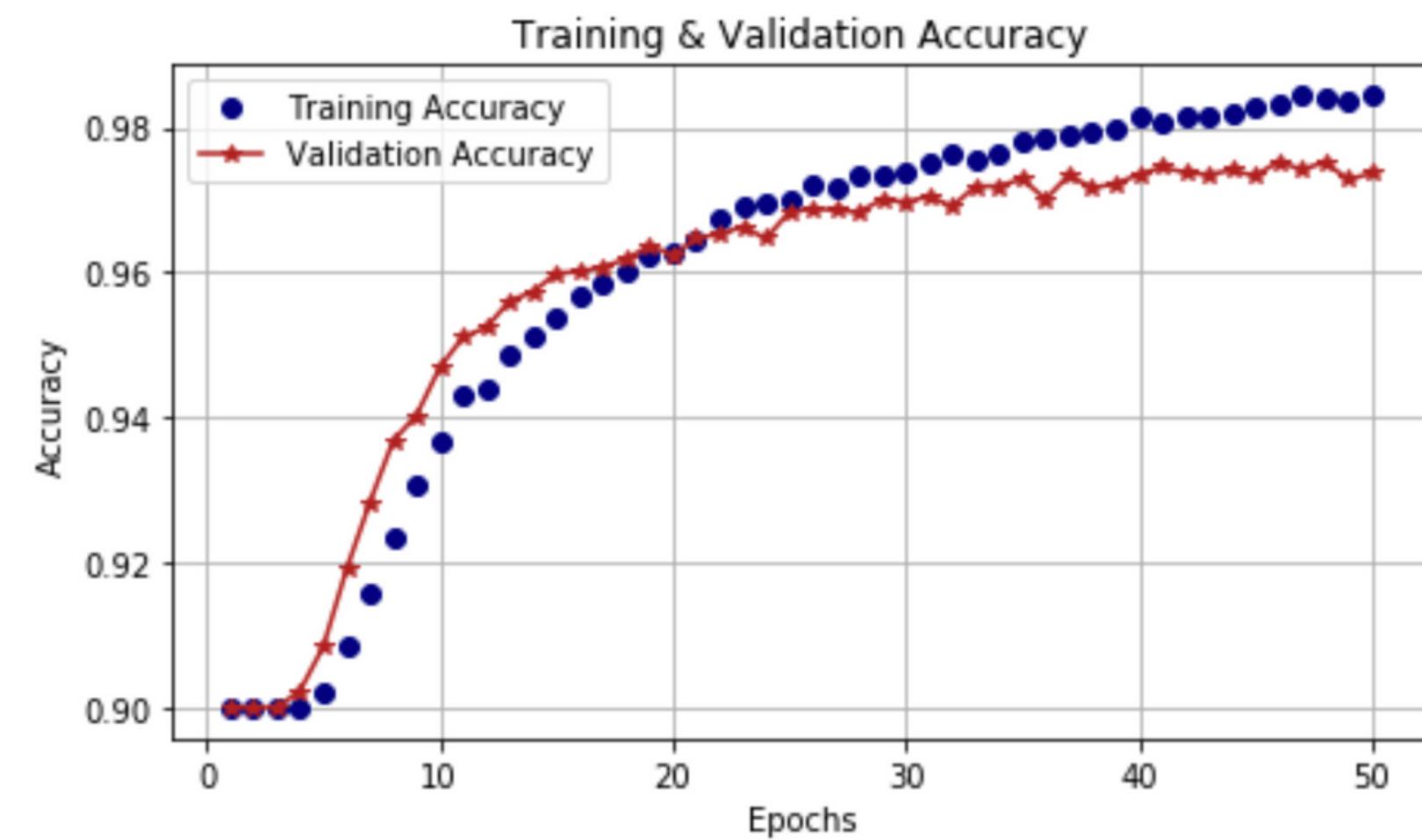
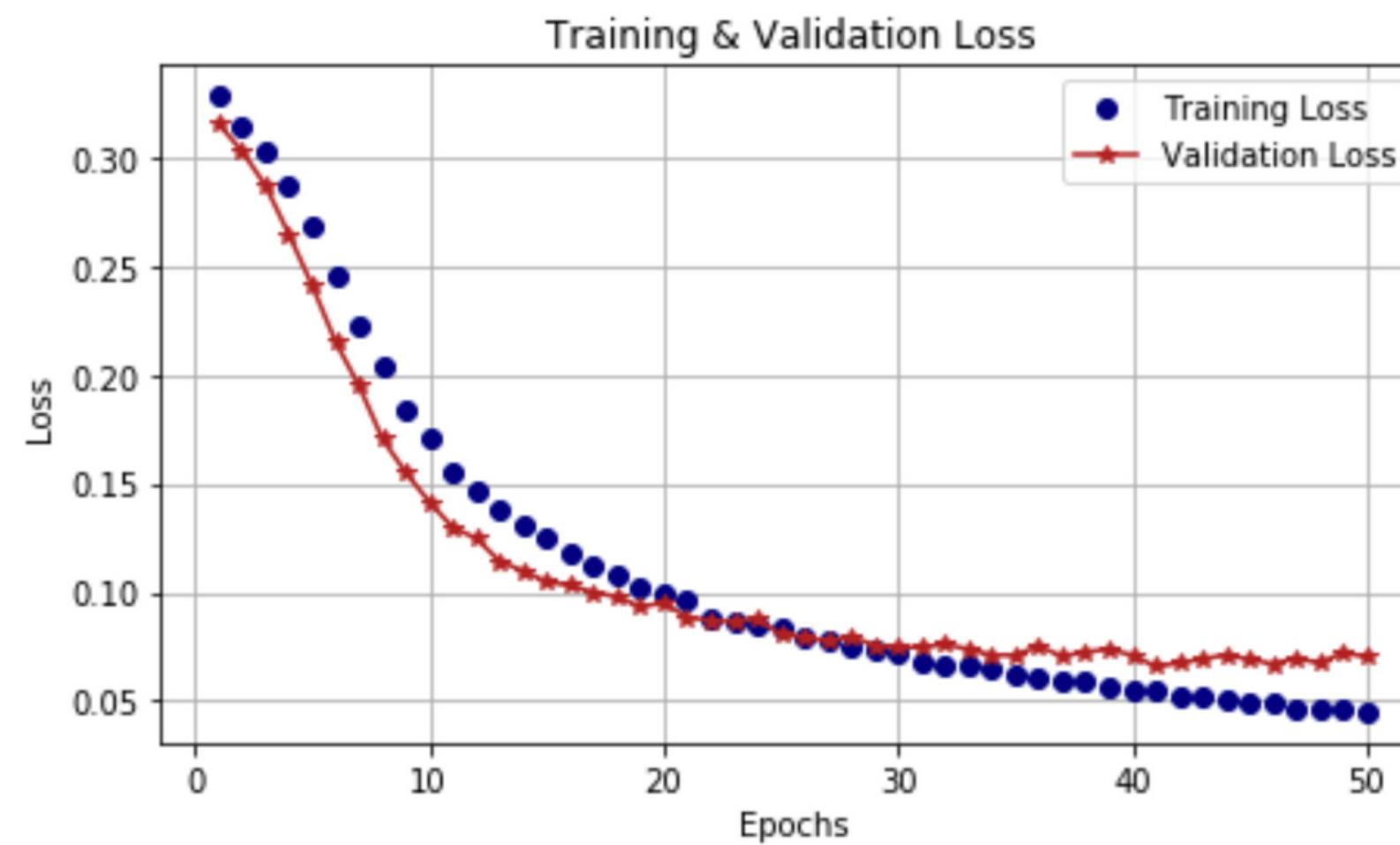
```
Epoch 1/50  
125/125 [=====] - 53s 426ms/step - loss: 0.3281 - accuracy: 0.9000 - val_loss: 0.3152 - val_accuracy: 0.9000  
Epoch 2/50  
125/125 [=====] - 48s 387ms/step - loss: 0.3143 - accuracy: 0.9000 - val_loss: 0.3028 - val_accuracy: 0.9000  
Epoch 3/50  
125/125 [=====] - 47s 377ms/step - loss: 0.3023 - accuracy: 0.9000 - val_loss: 0.2869 - val_accuracy: 0.9000  
Epoch 4/50  
125/125 [=====] - 47s 373ms/step - loss: 0.2867 - accuracy: 0.9001 - val_loss: 0.2645 - val_accuracy: 0.9023  
Epoch 5/50  
125/125 [=====] - 46s 372ms/step - loss: 0.2680 - accuracy: 0.9023 - val_loss: 0.2409 - val_accuracy: 0.9084  
. . .  
Epoch 49/50  
125/125 [=====] - 46s 368ms/step - loss: 0.0459 - accuracy: 0.9837 - val_loss: 0.0722 - val_accuracy: 0.9728  
Epoch 50/50  
125/125 [=====] - 47s 372ms/step - loss: 0.0450 - accuracy: 0.9844 - val_loss: 0.0702 - val_accuracy: 0.9737
```

Lama pemrosesan : ± 40 menit

TRAINING OUTPUT

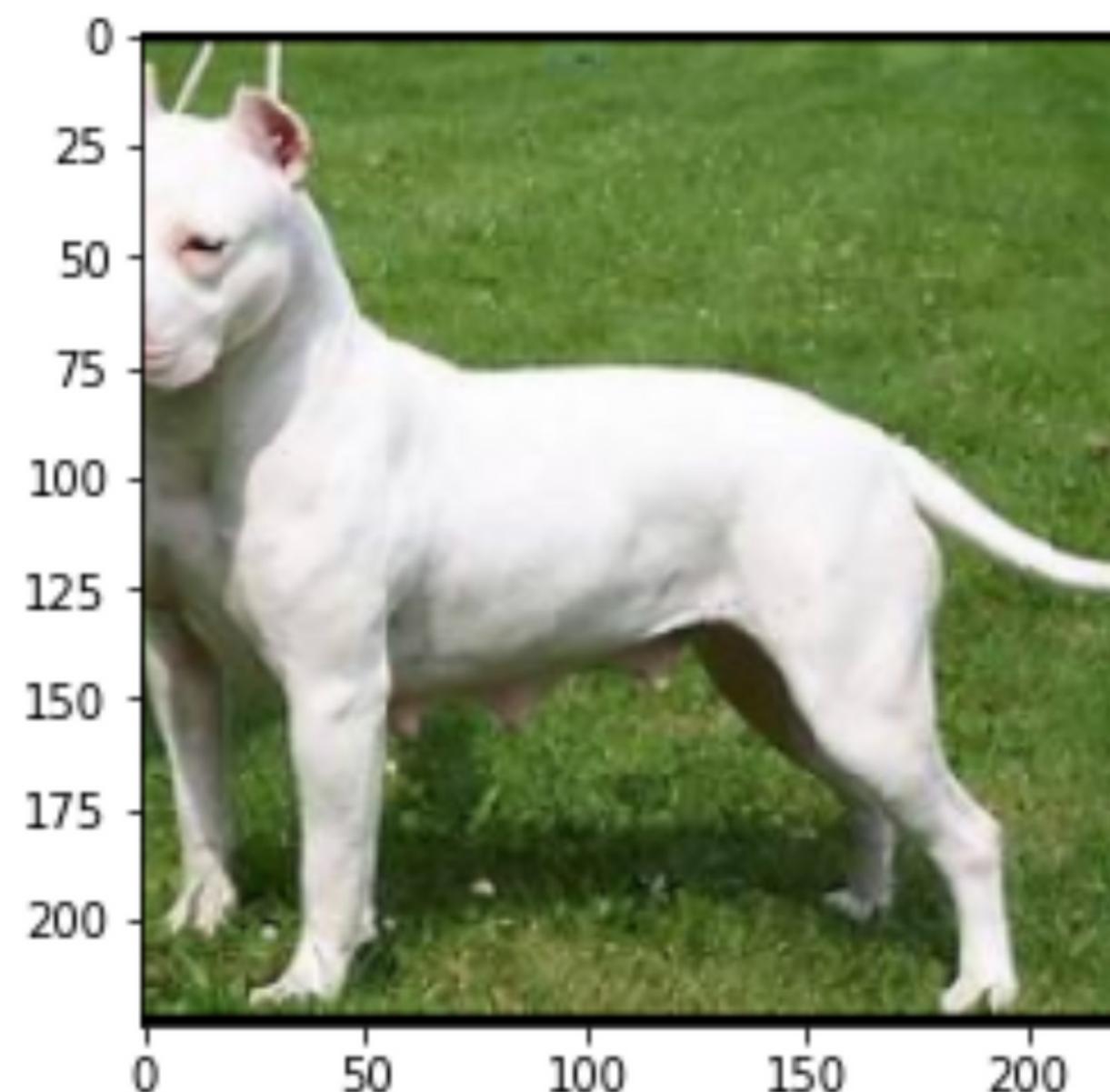
CNN : Epochs=50, Train accuracy=0.98438, Validation accuracy=0.97370

ACCURACY

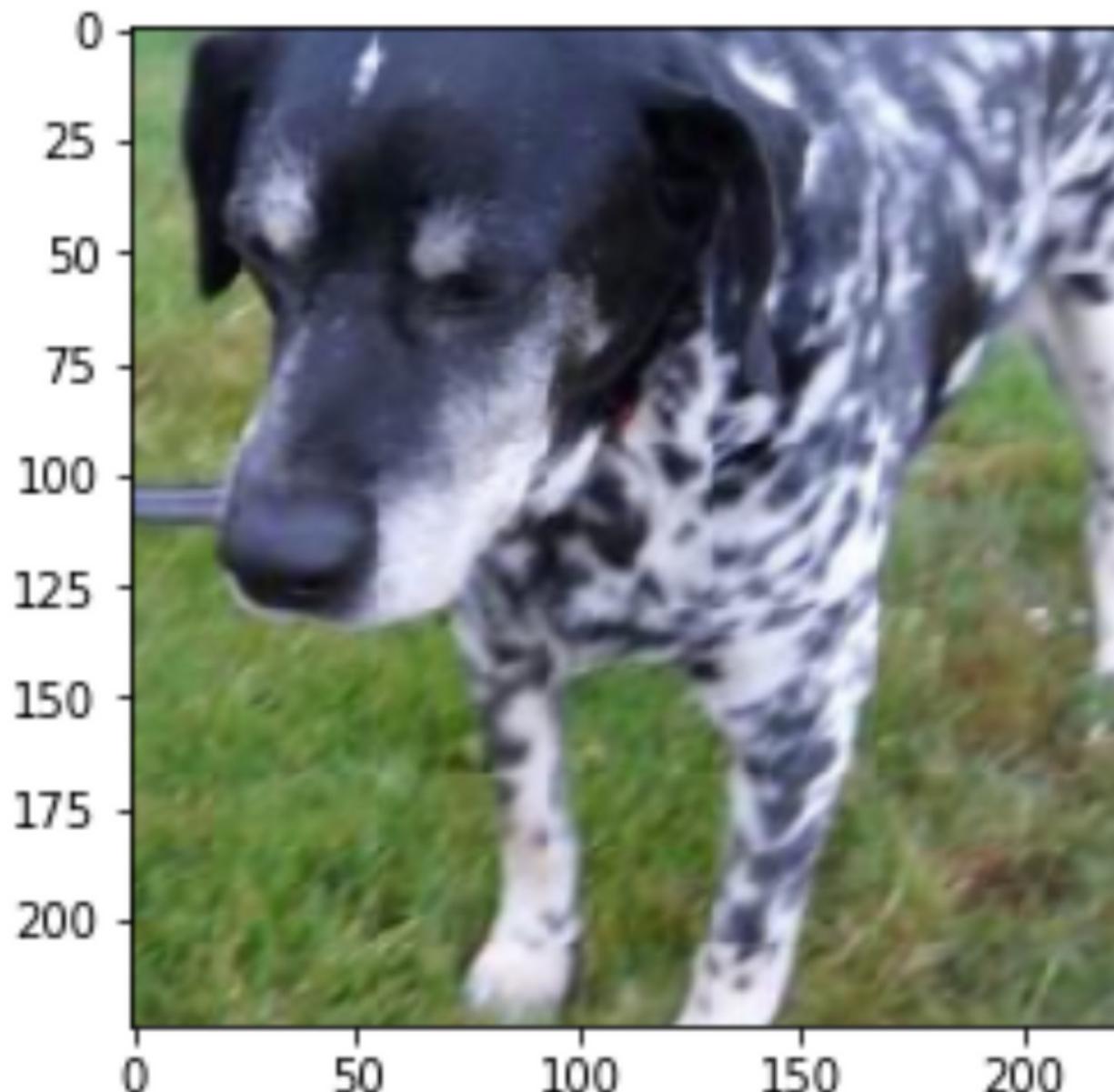


Manual Testing

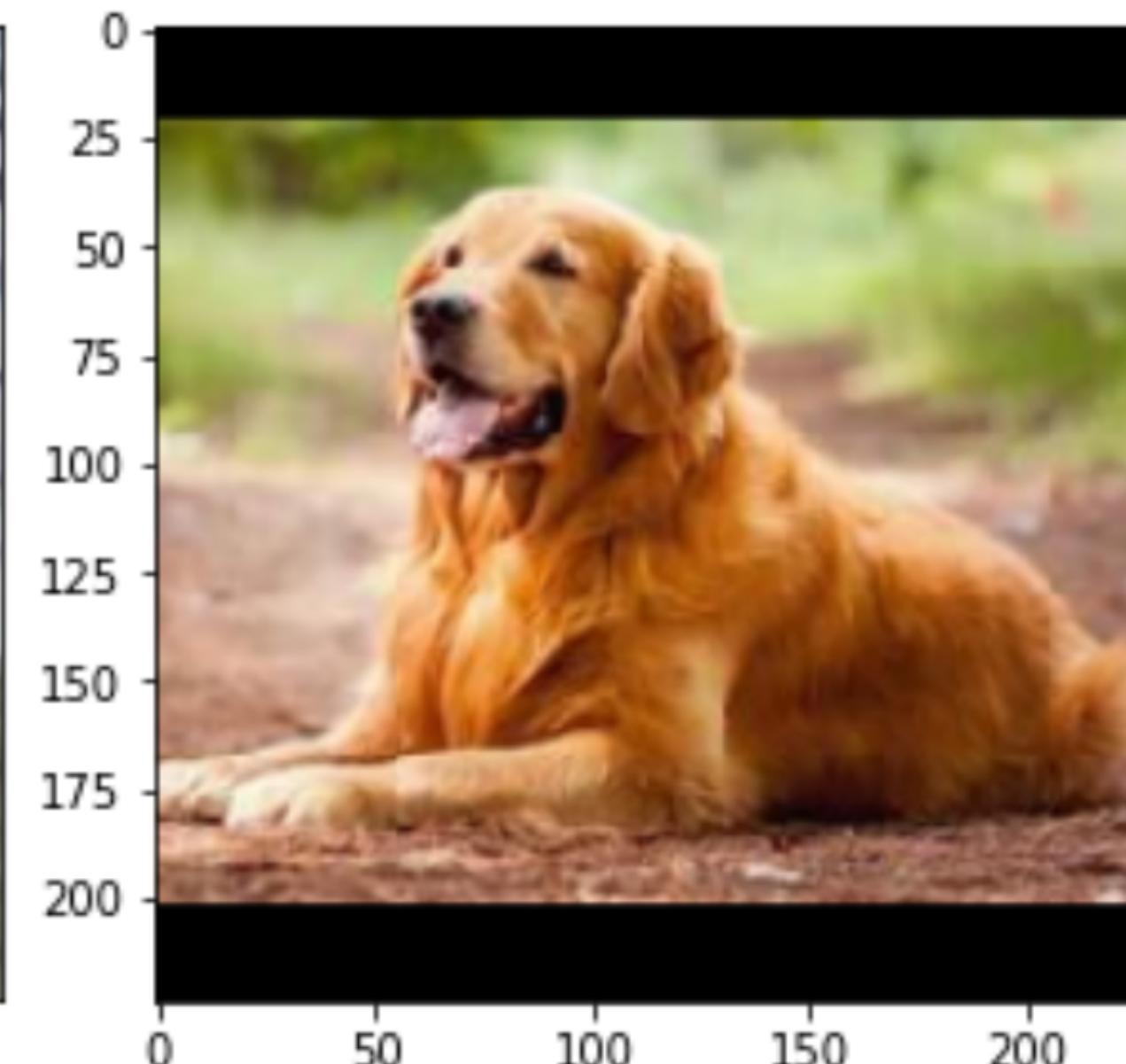




Actual : anjing
Predict : anjing



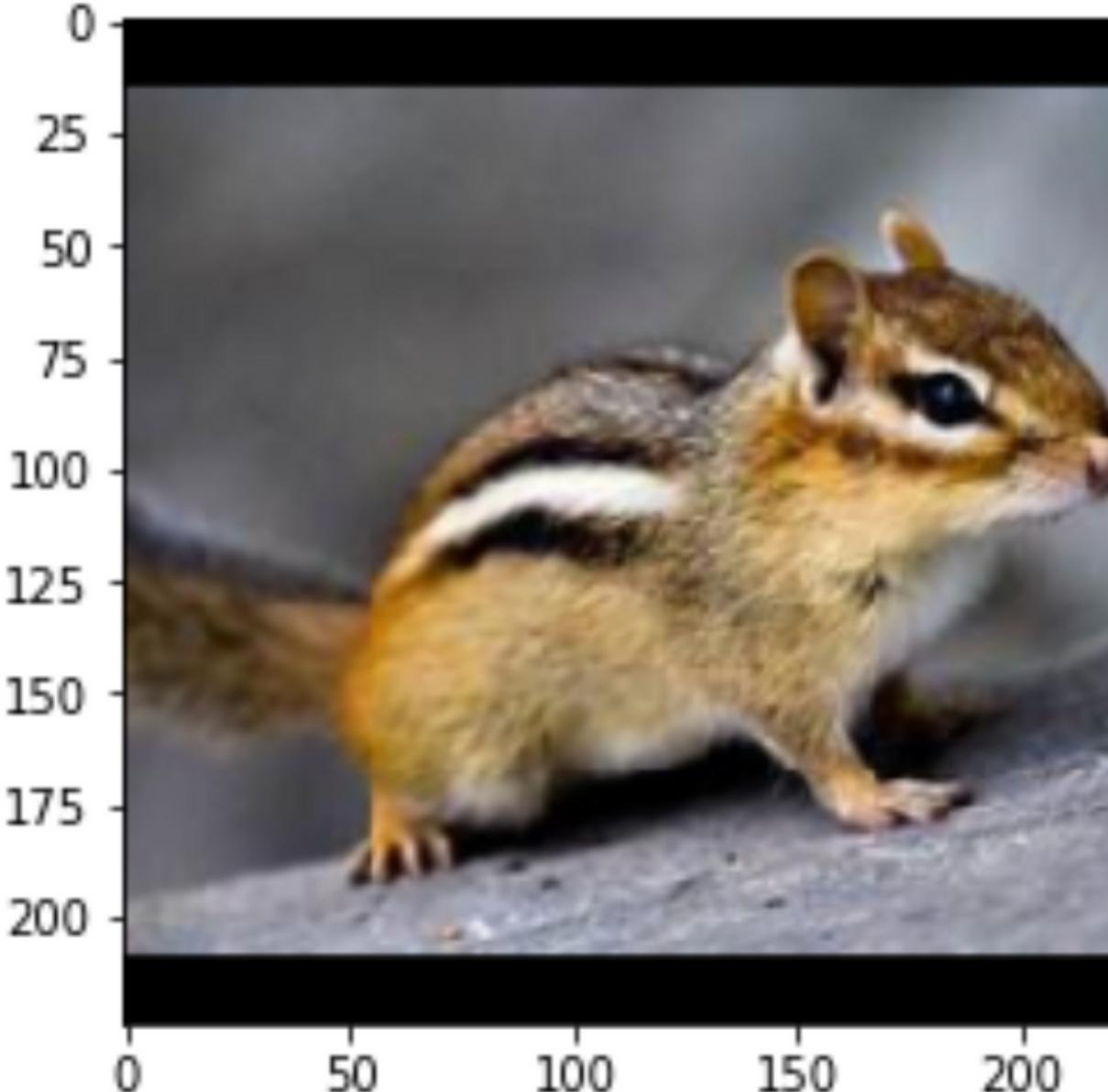
Actual : anjing
Predict : anjing



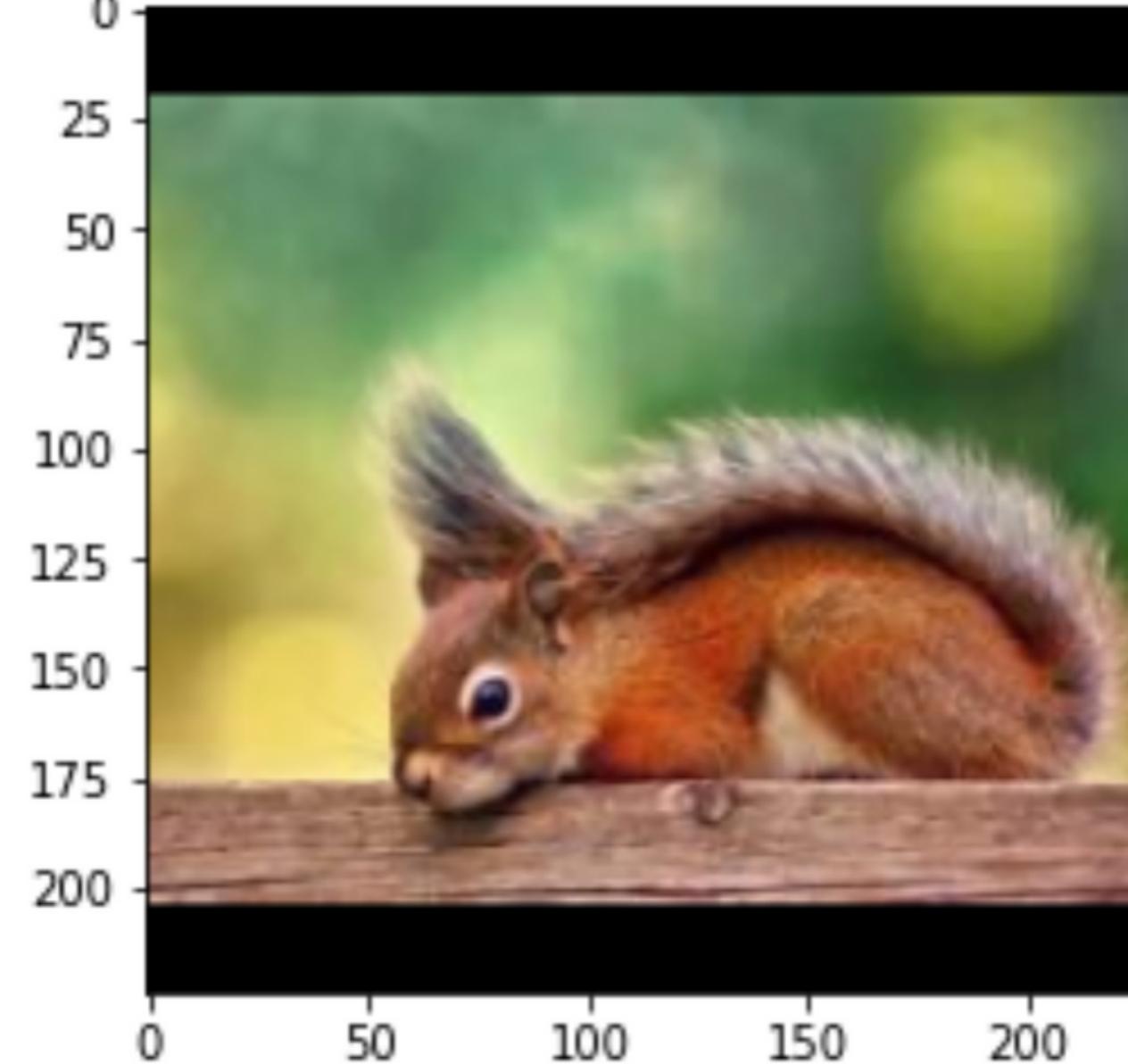
Actual : anjing
Predict : anjing



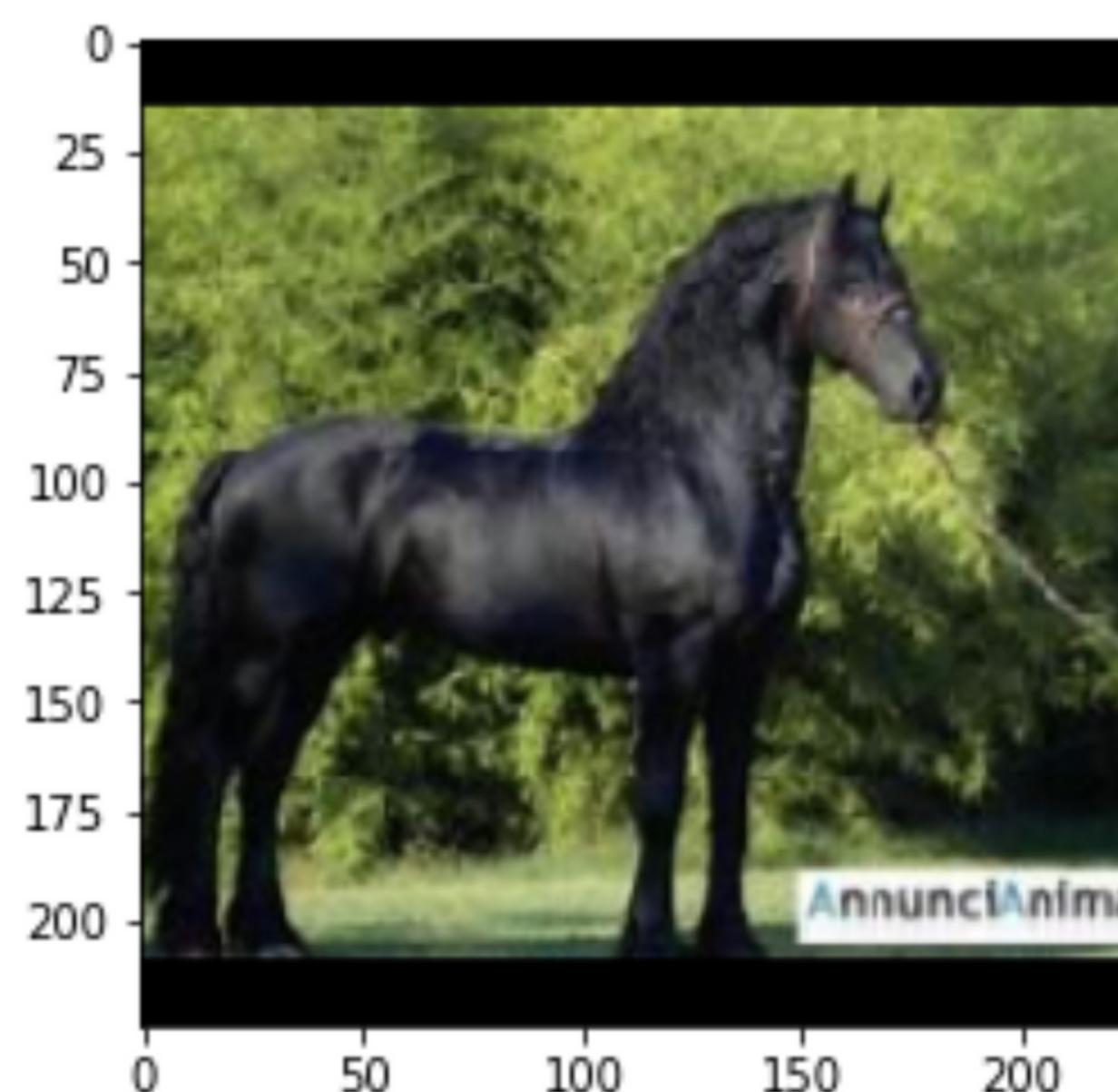
Actual : tupai
Predict : tupai



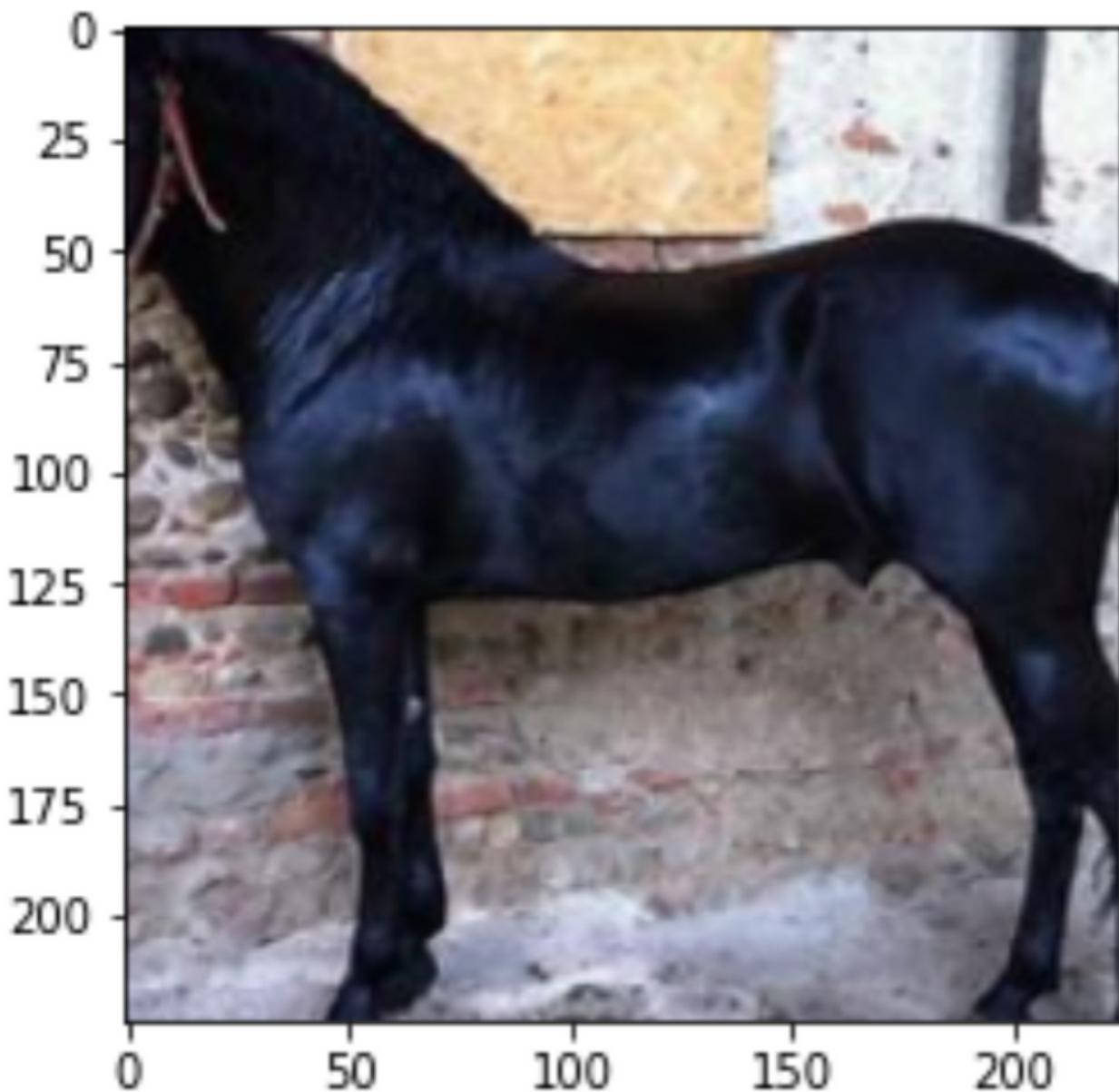
Actual : tupai
Predict : tupai



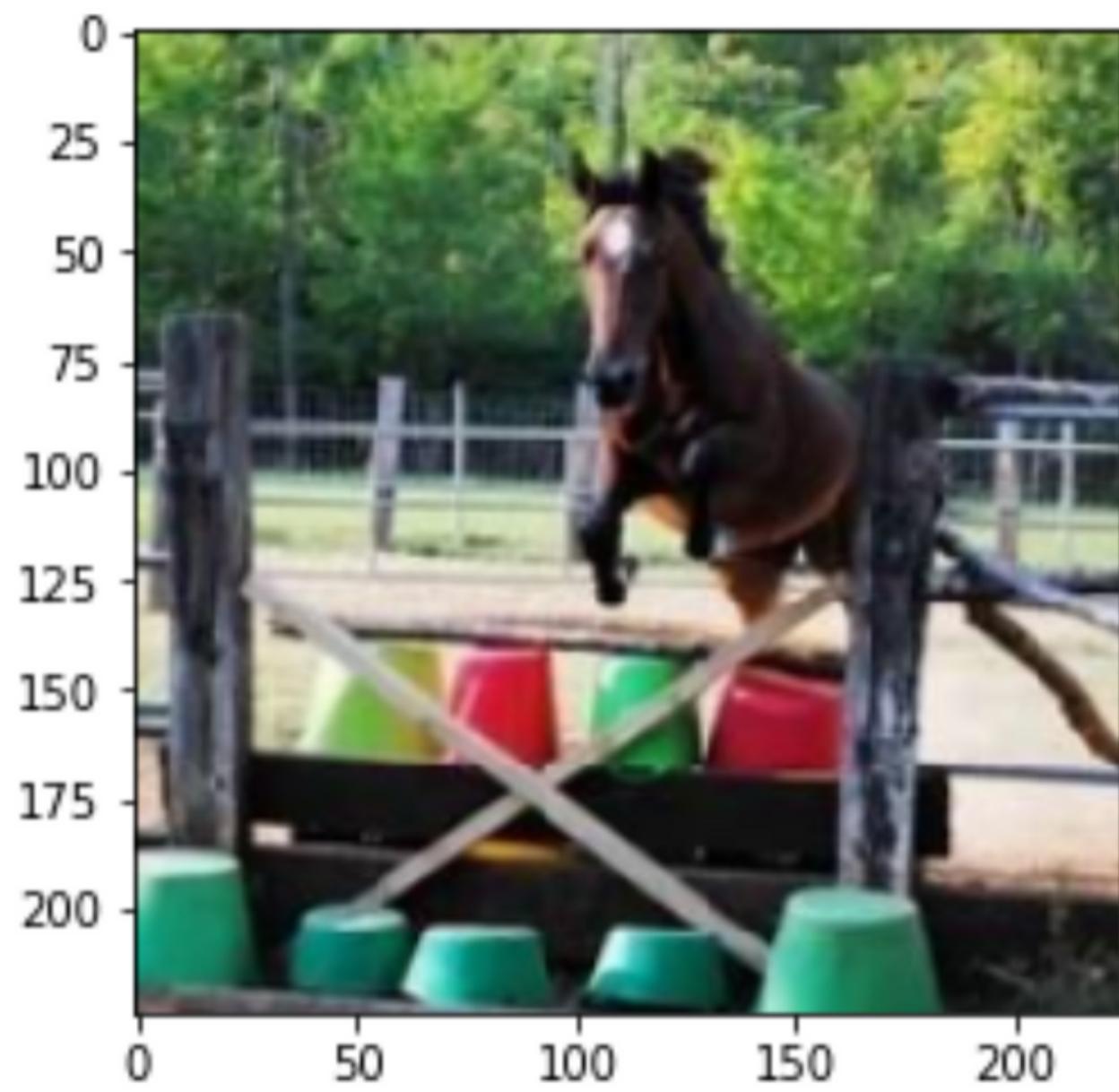
Actual : tupai
Predict : tupai



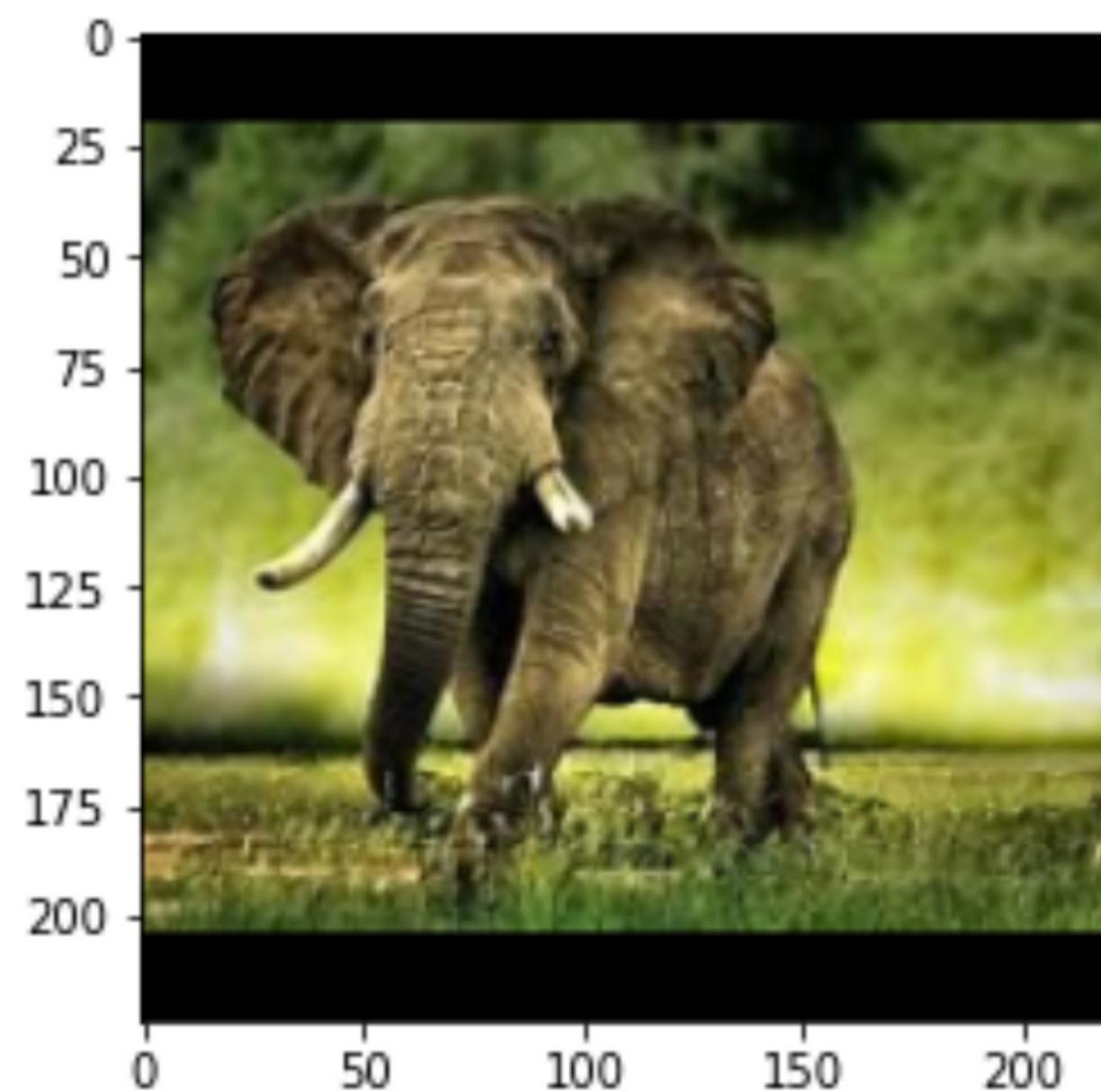
Actual : kuda
Predict : kuda



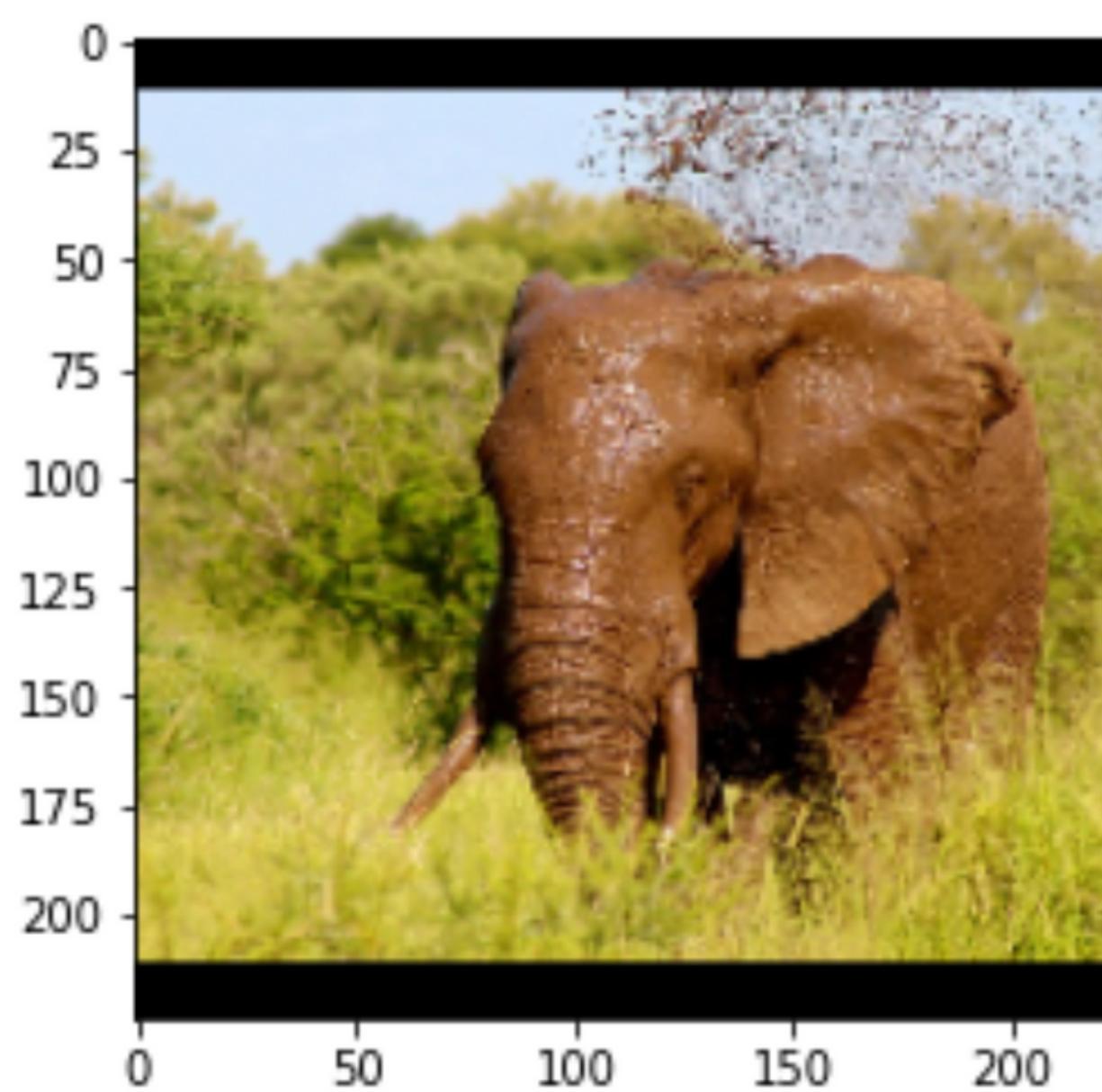
Actual : kuda
Predict : kuda



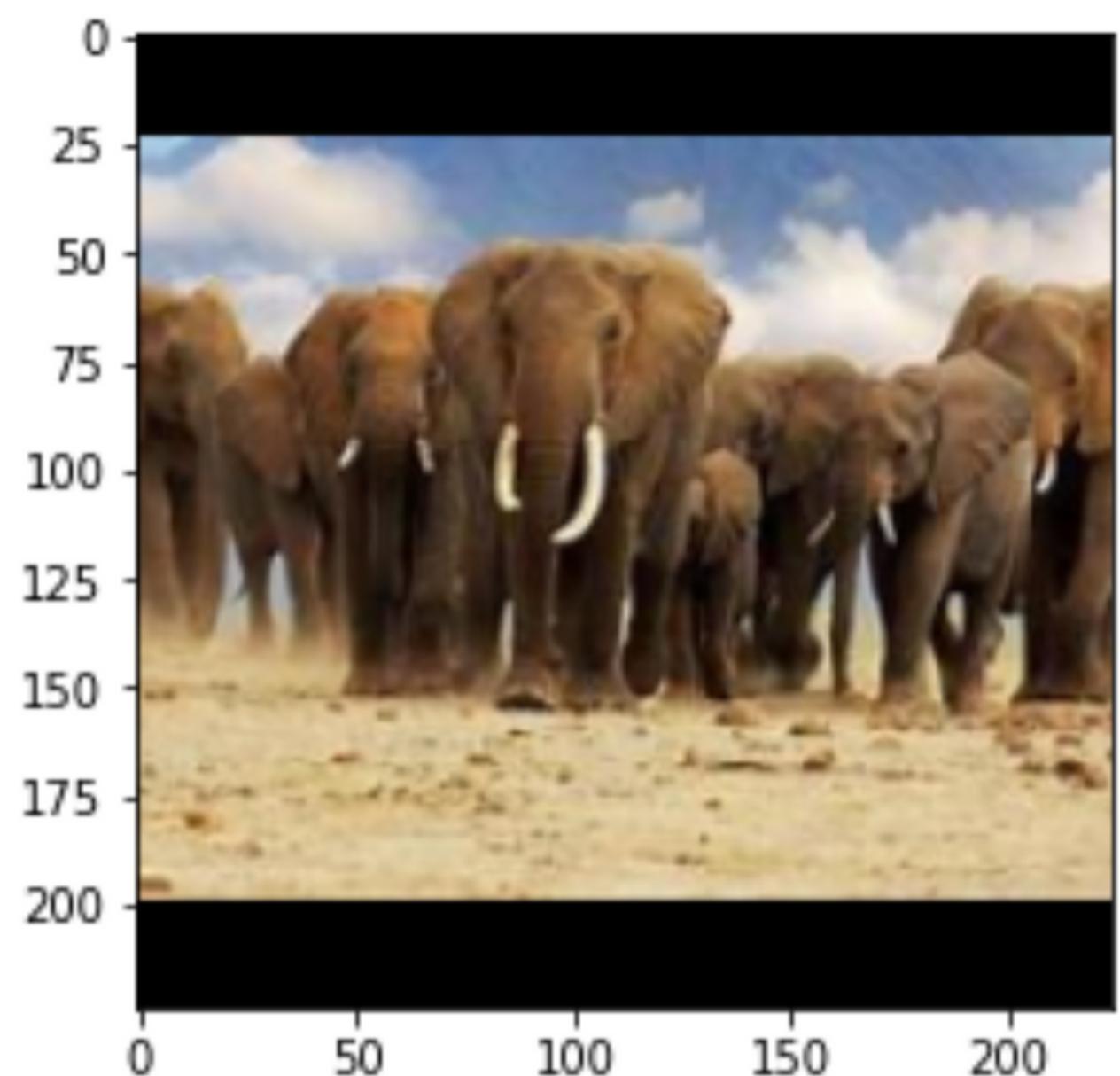
Actual : kuda
Predict : kuda



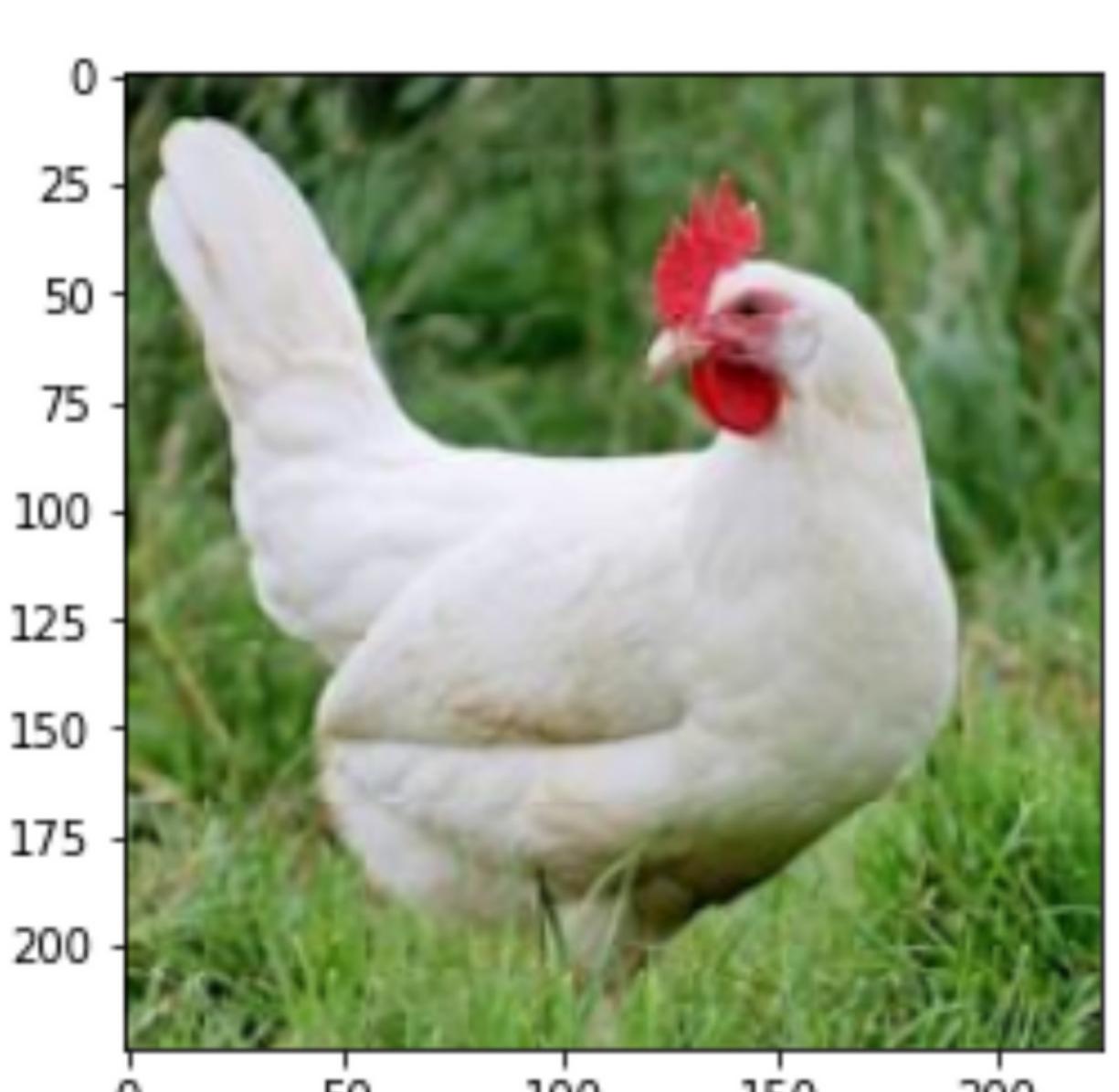
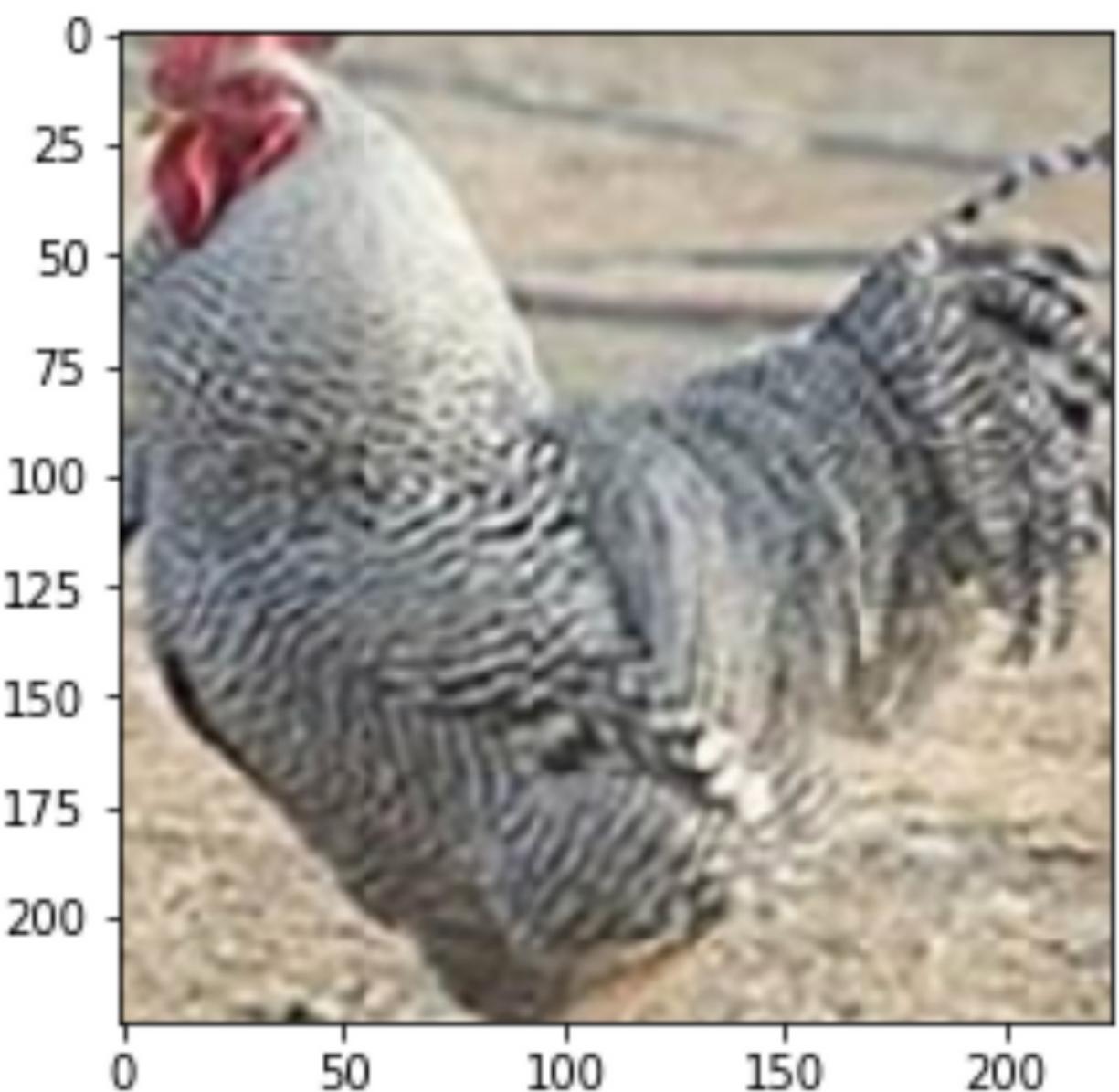
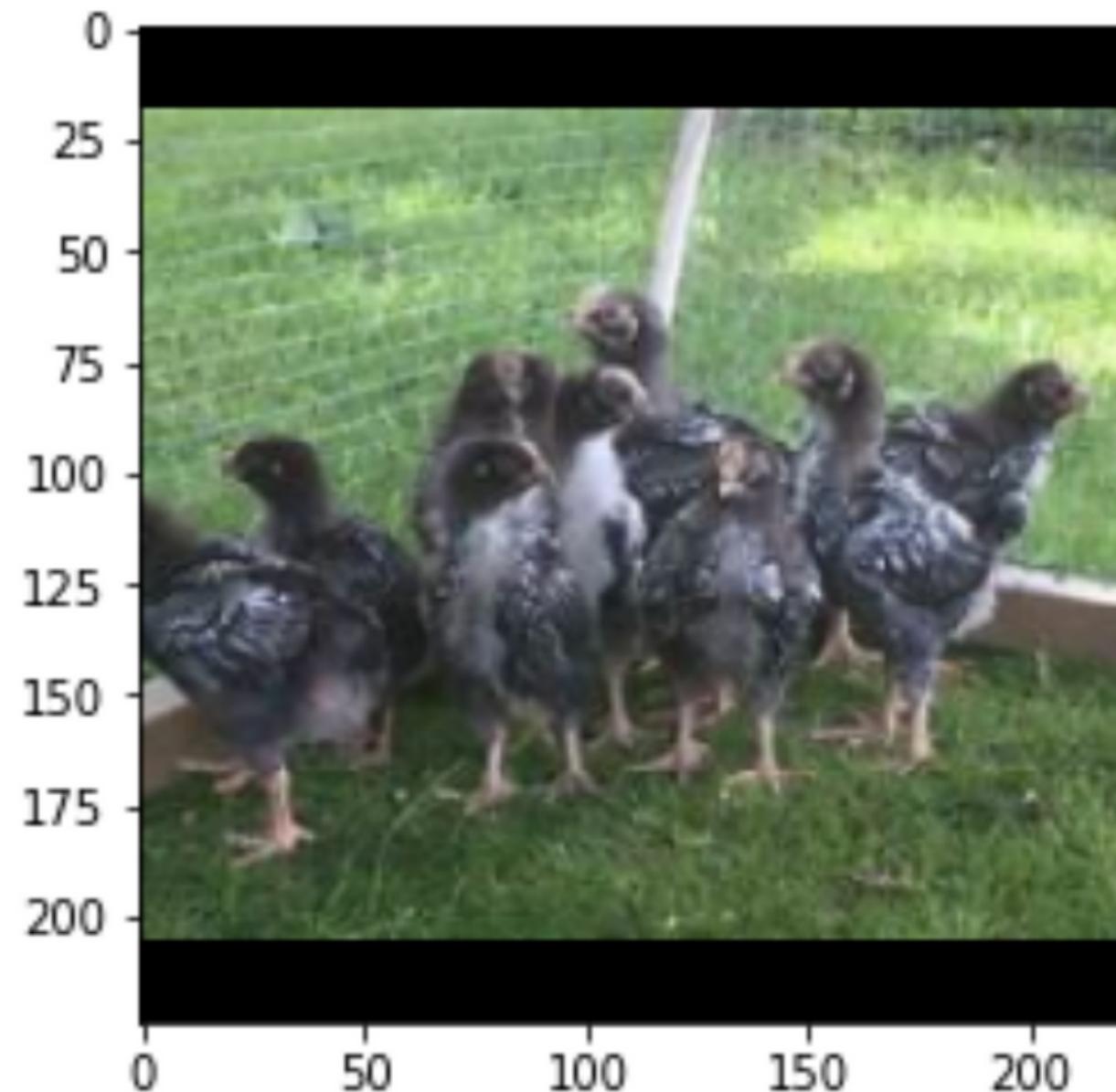
Actual : gajah
Predict : gajah



Actual : gajah
Predict : gajah



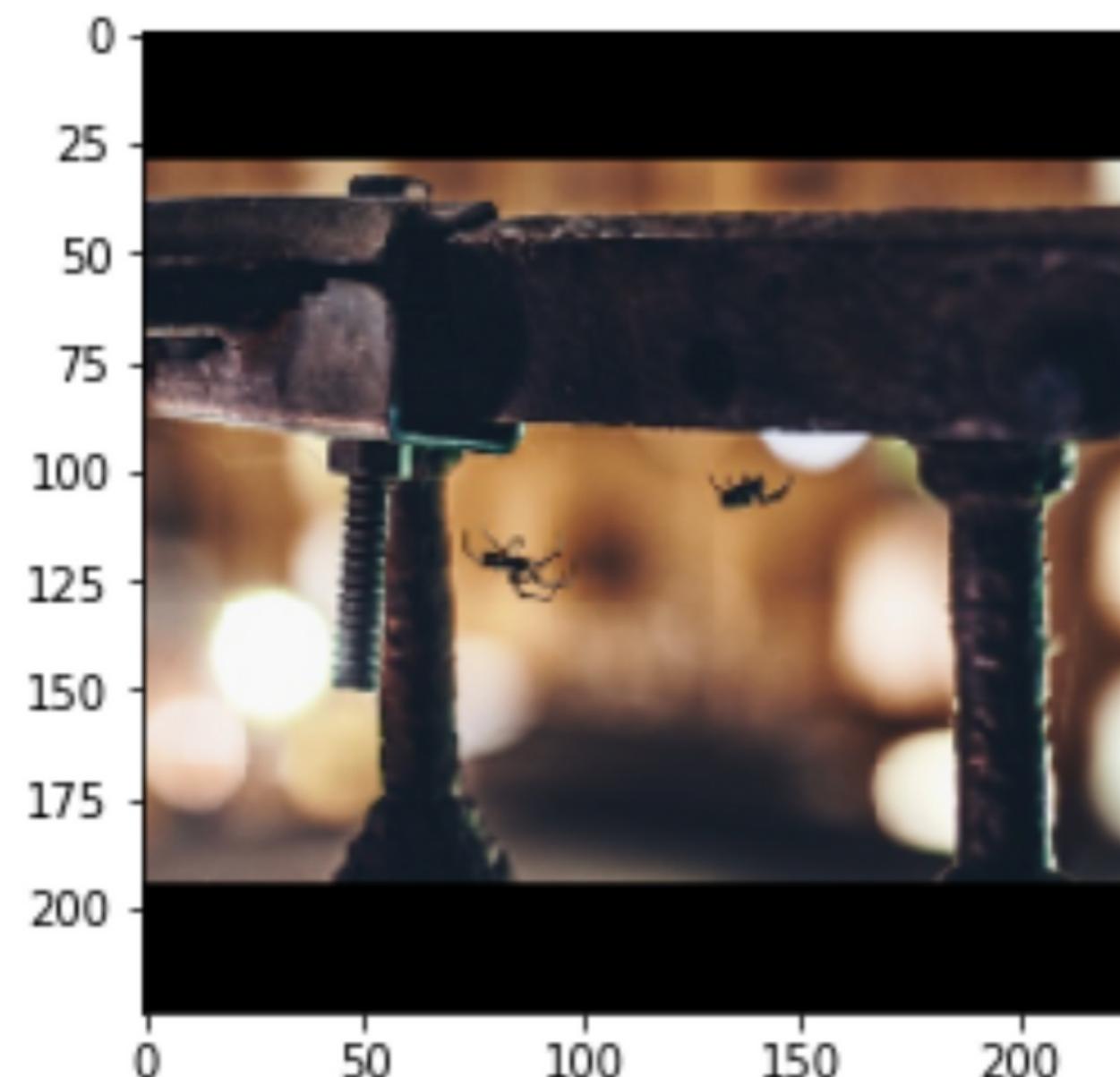
Actual : gajah
Predict : gajah



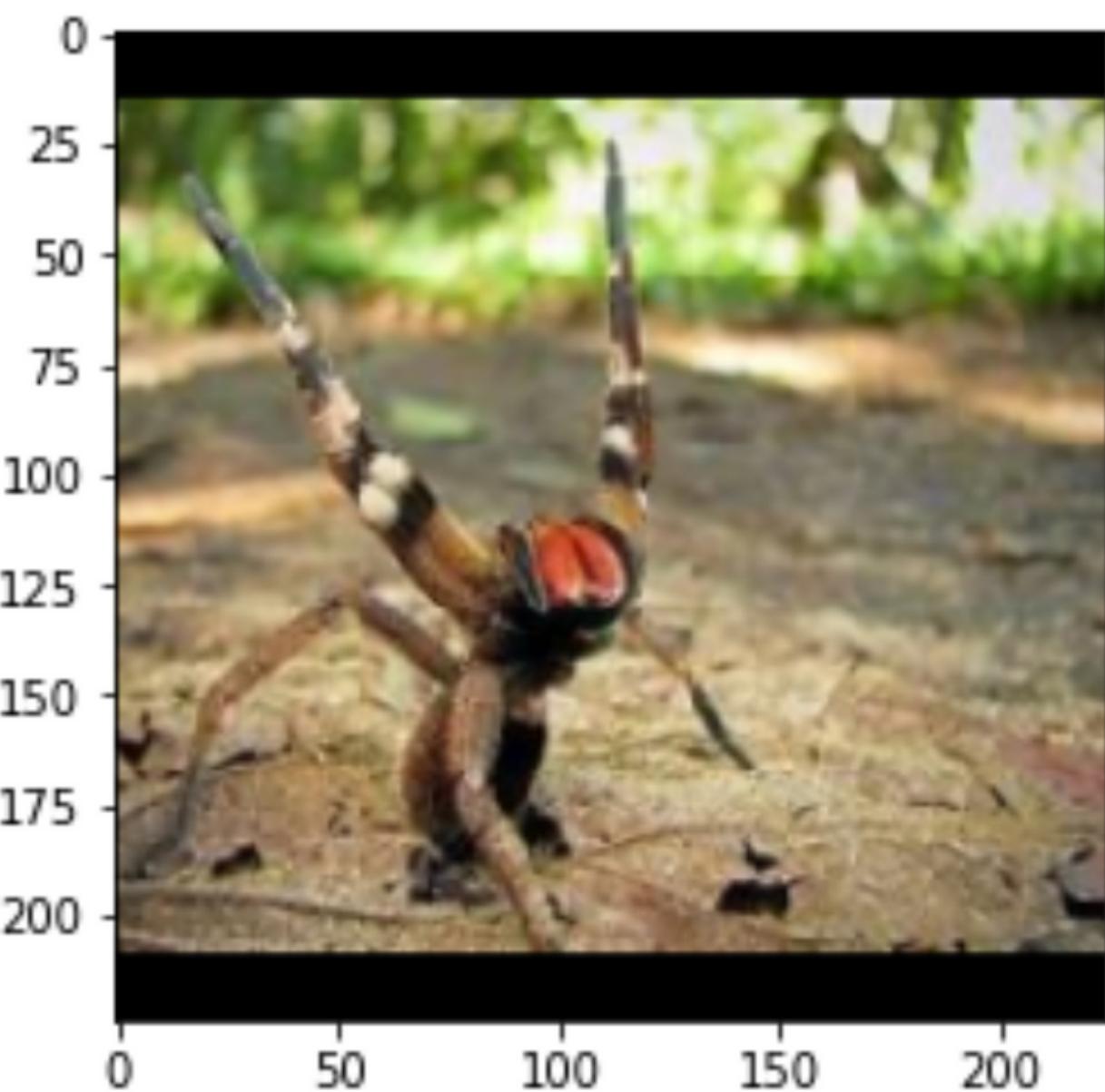
Actual : ayam
Predict : domba

Actual : ayam
Predict : ayam

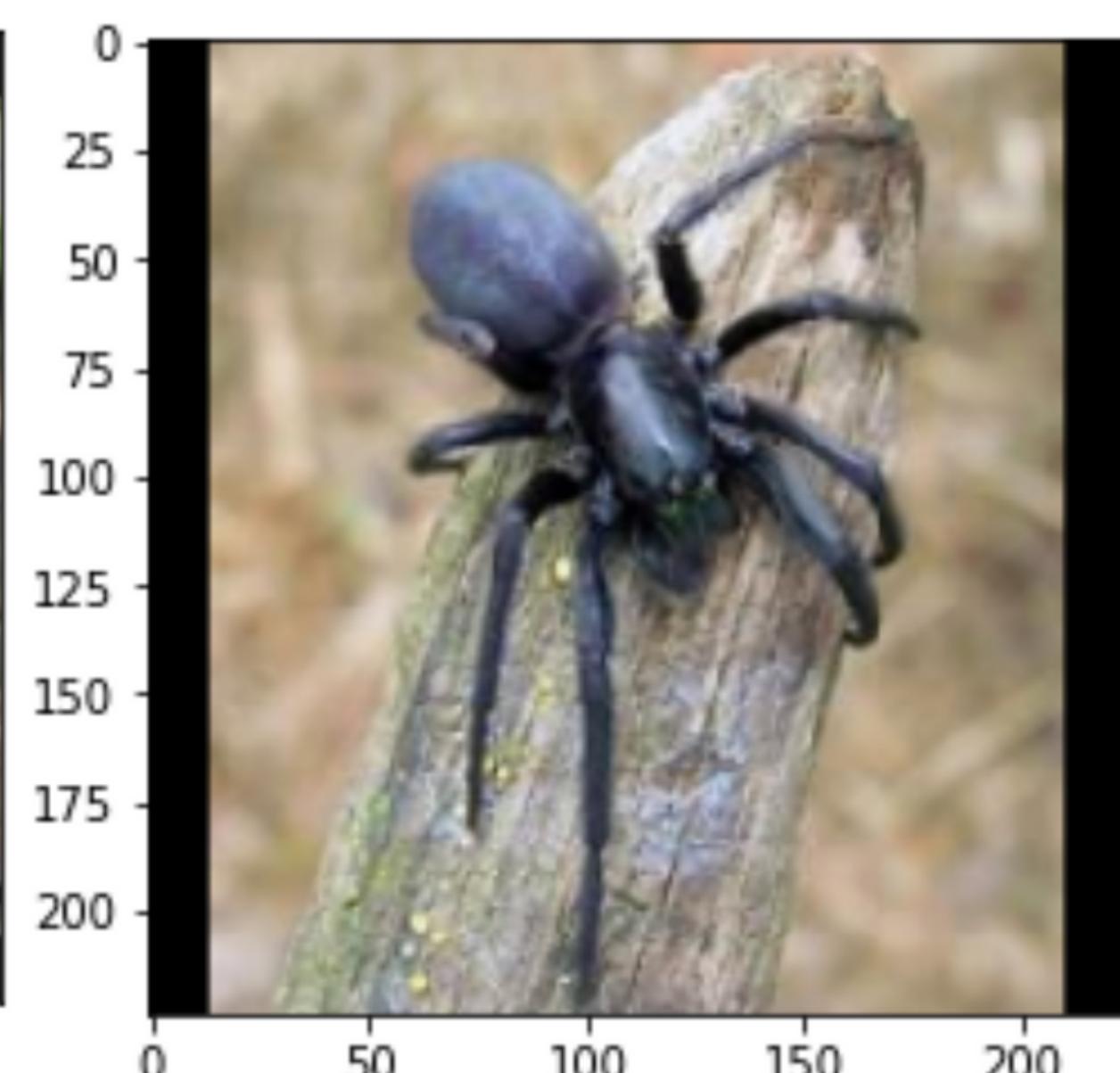
Actual : ayam
Predict : ayam



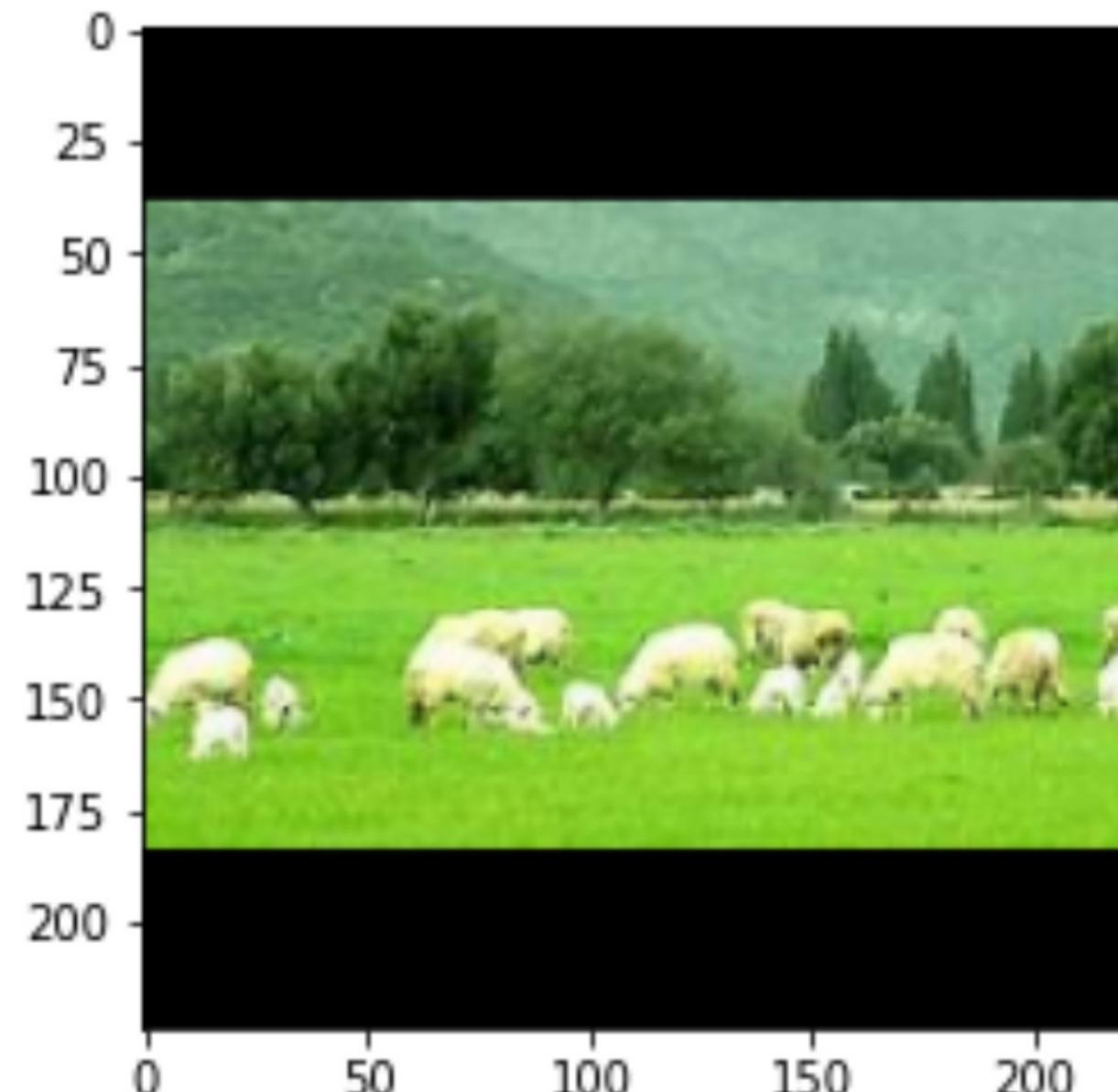
Actual : laba-laba
Predict : kupu-kupu



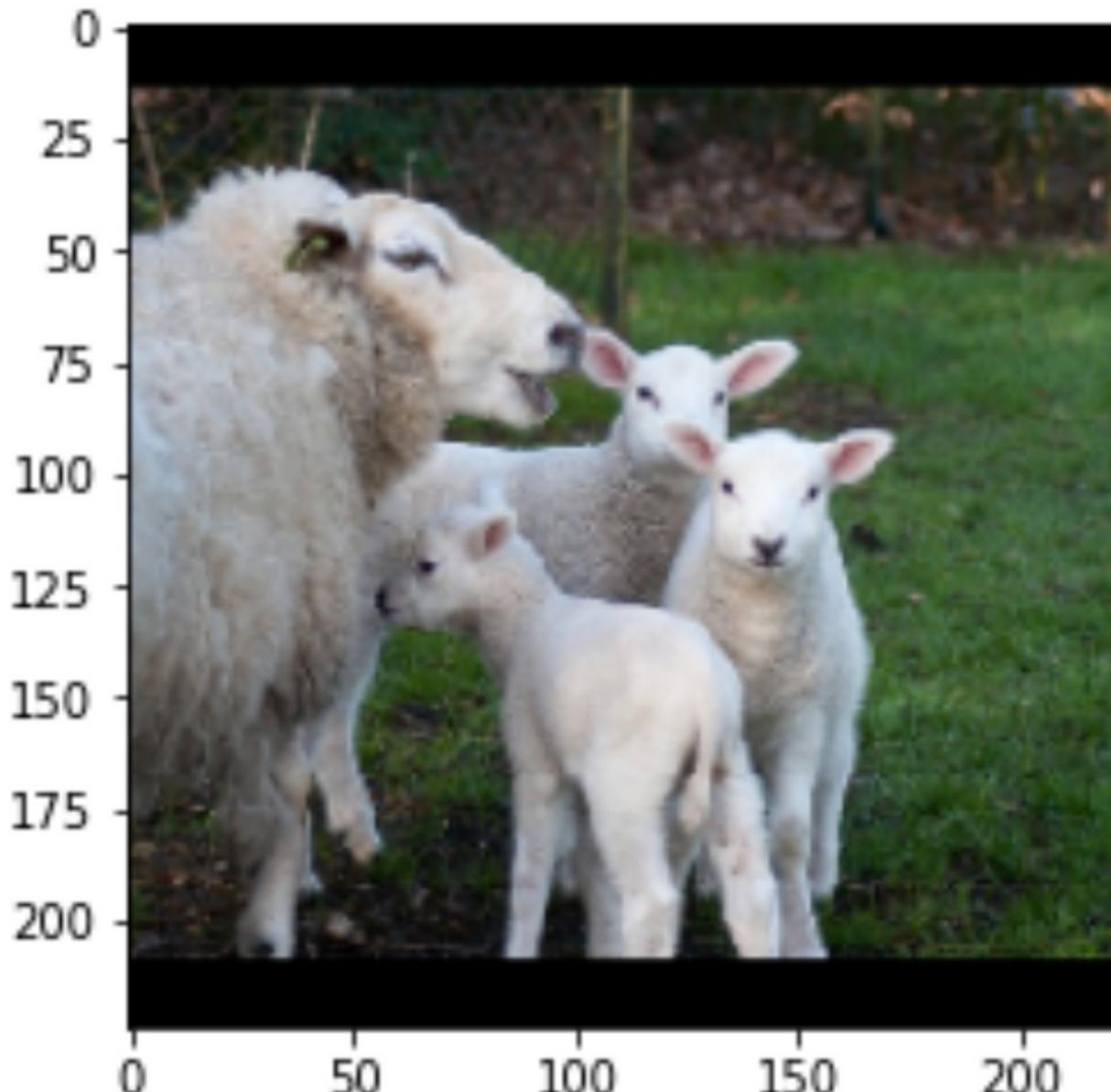
Actual : laba-laba
Predict : laba-laba



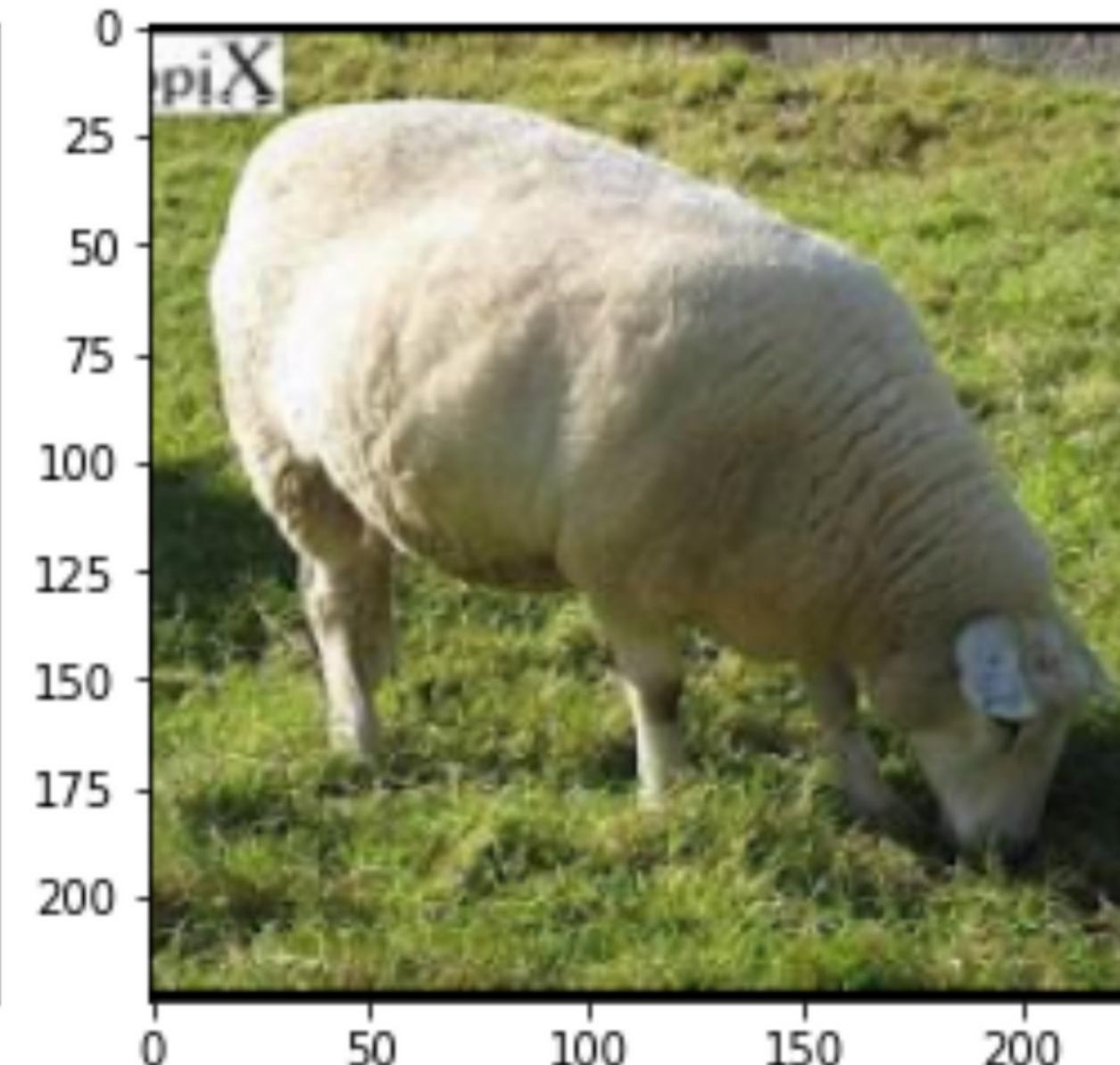
Actual : laba-laba
Predict : laba-laba



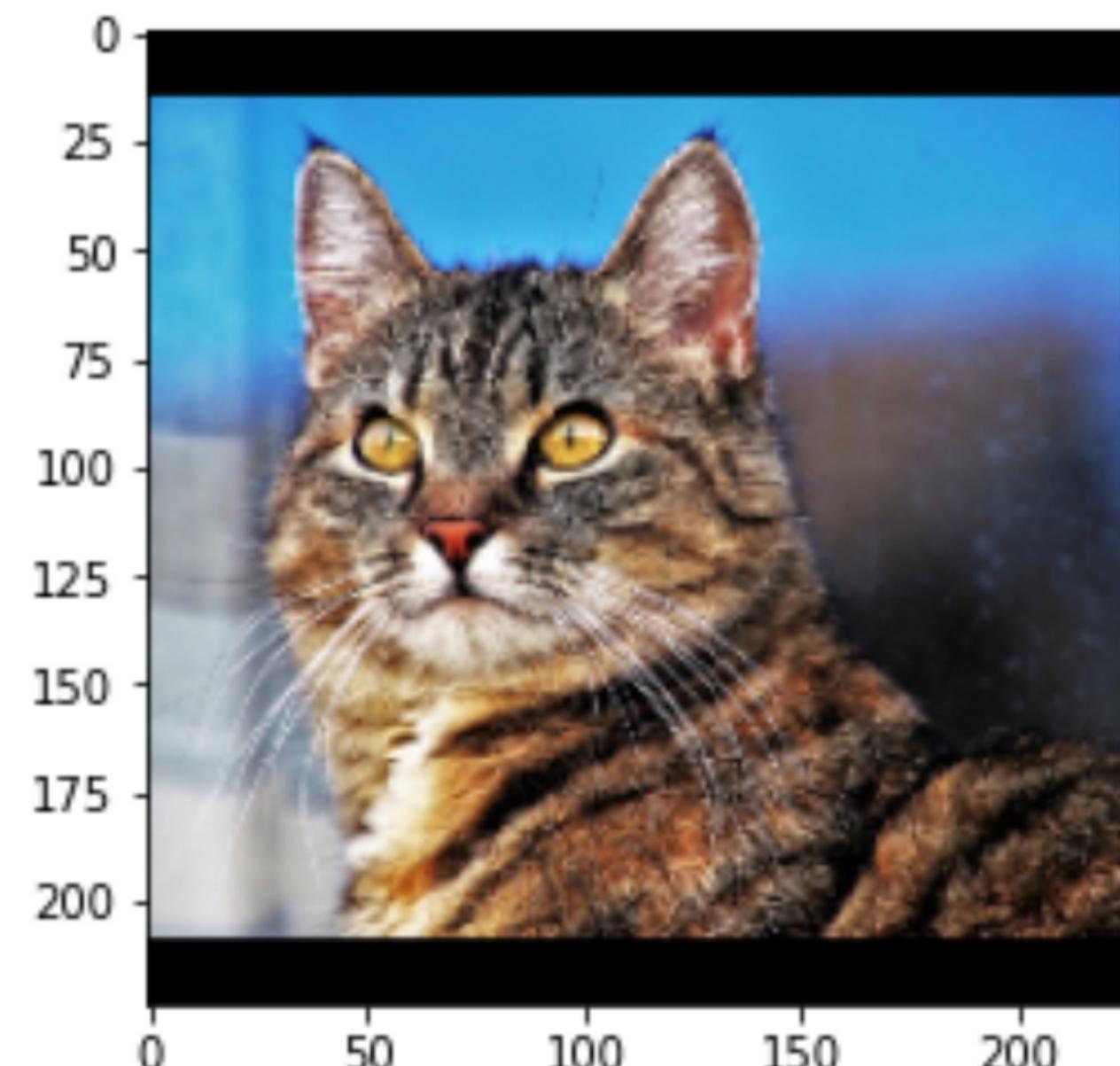
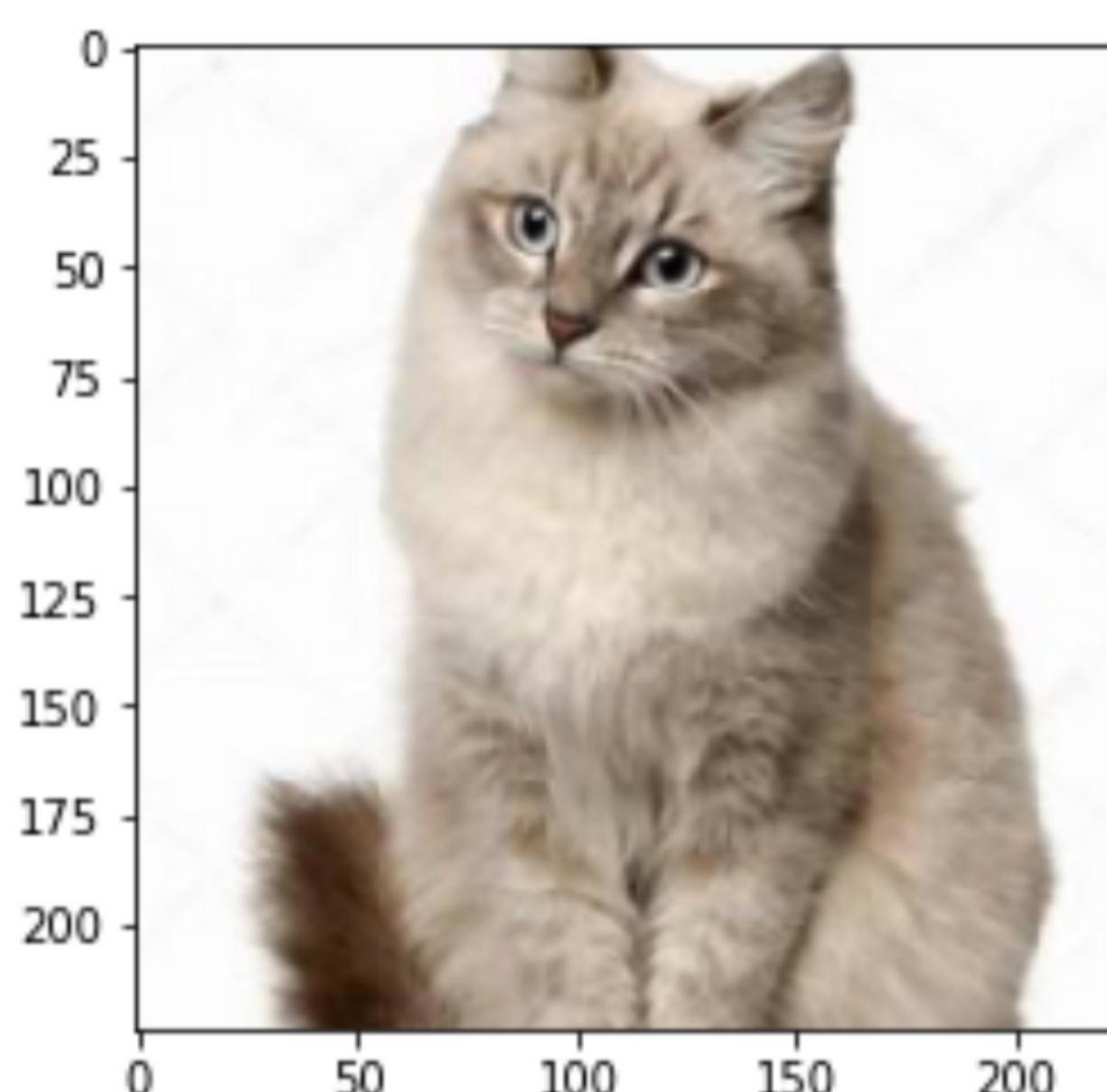
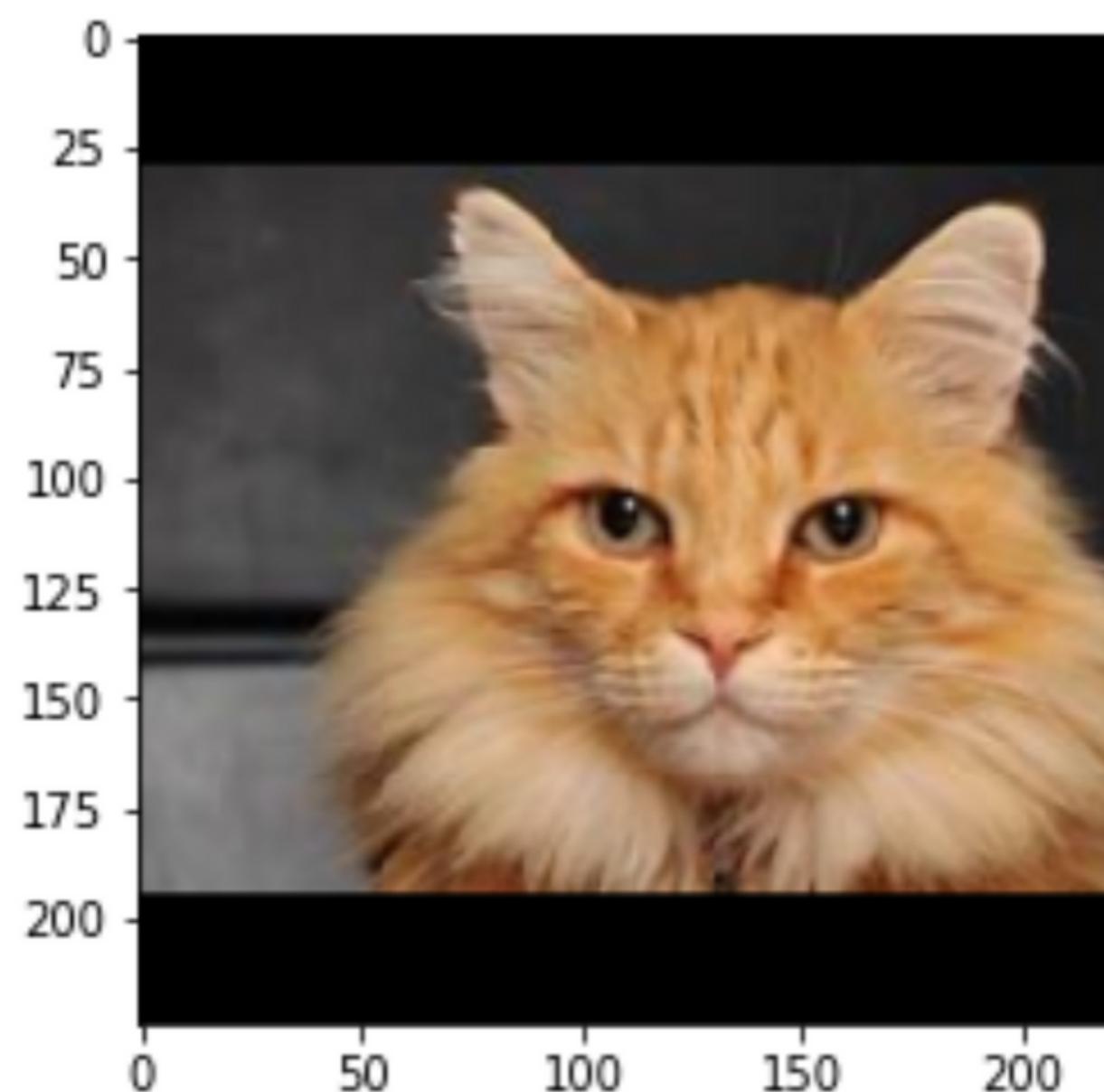
Actual : domba
Predict : domba



Actual : domba
Predict : anjing



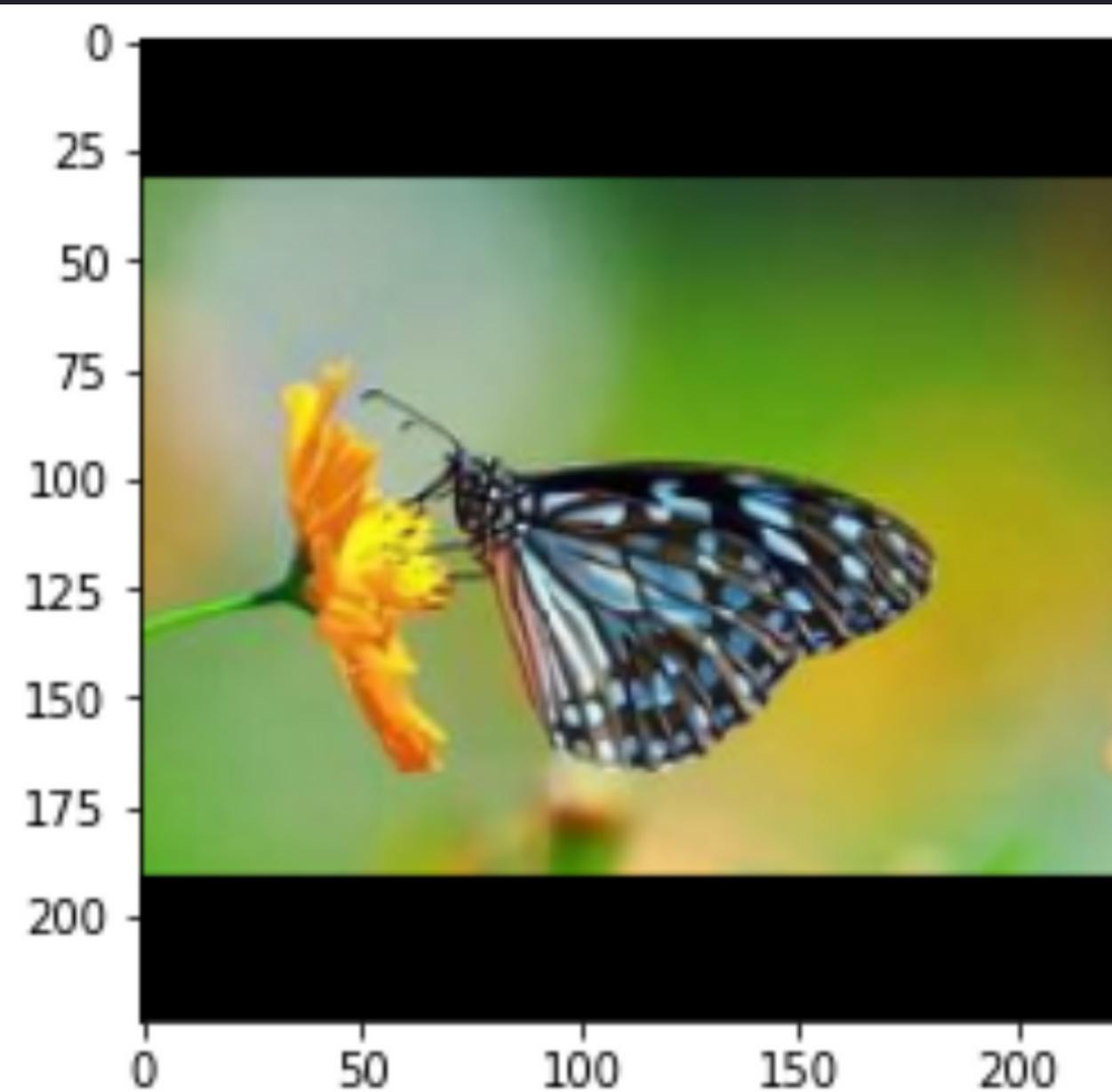
Actual : domba
Predict : domba



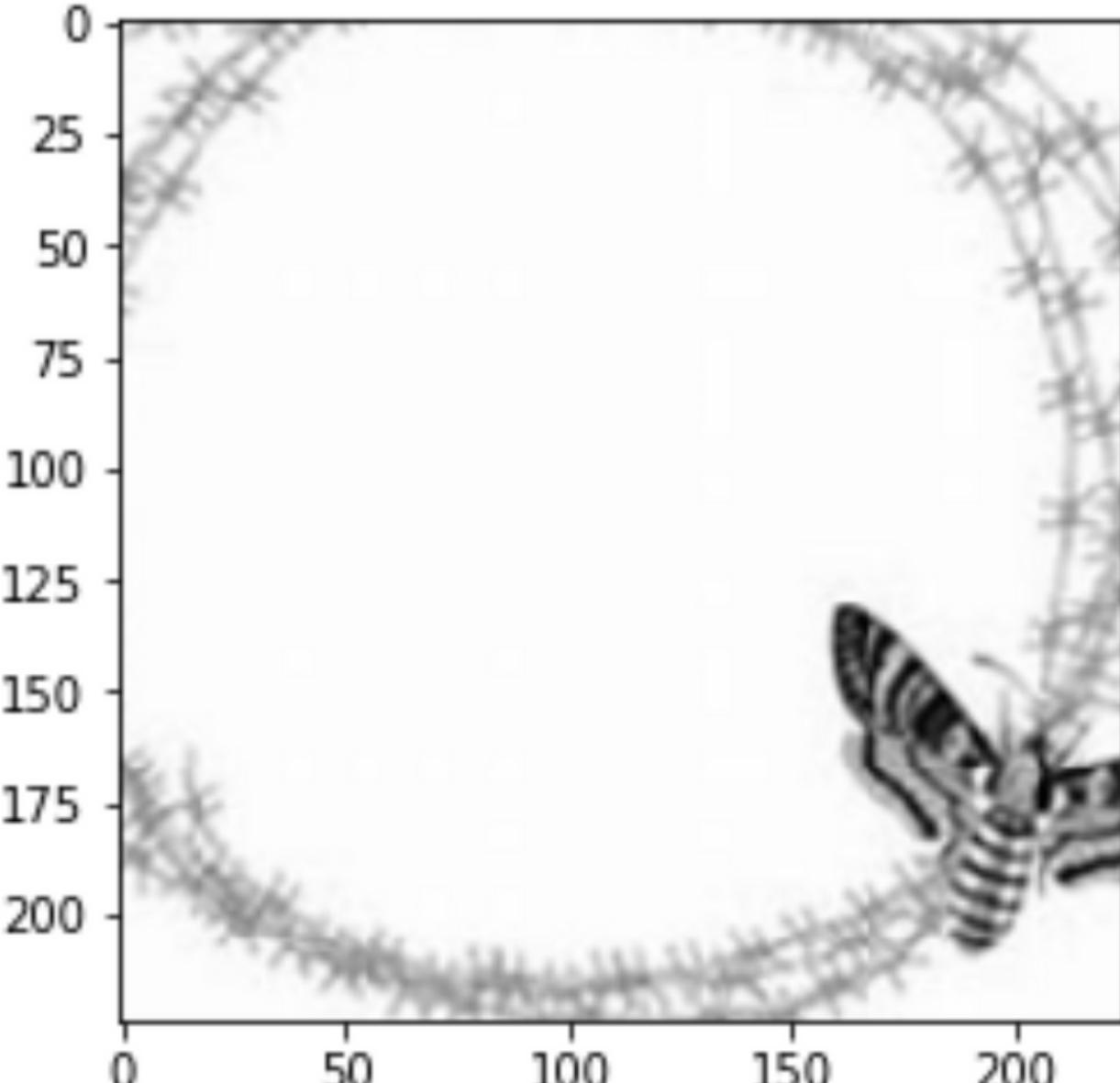
Actual : kucing
Predict : kucing

Actual : kucing
Predict : kucing

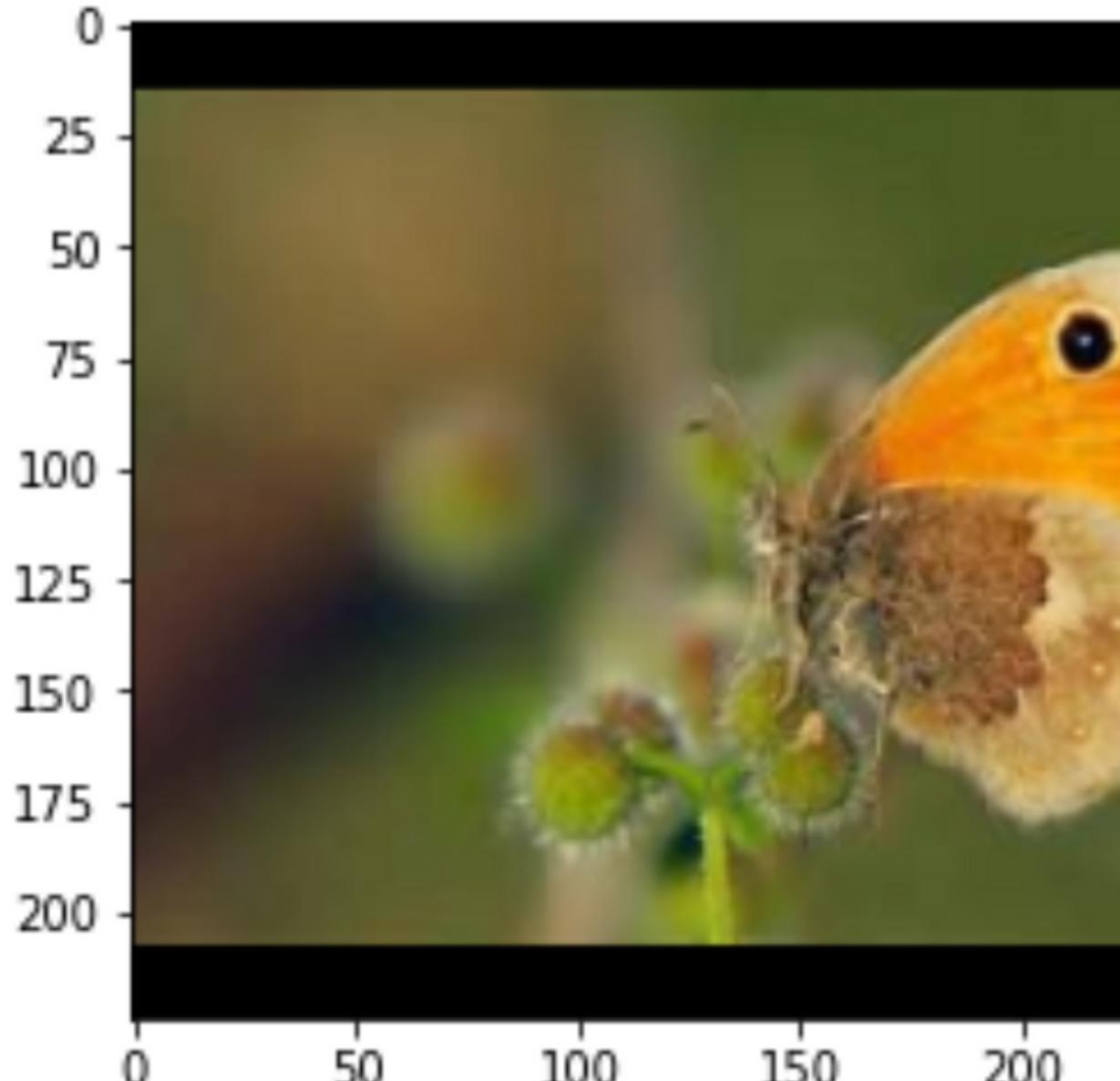
Actual : kucing
Predict : kucing



Actual : kupu-kupu
Predict : kupu-kupu



Actual : kupu-kupu
Predict : kupu-kupu



Actual : kupu-kupu
Predict : kupu-kupu



SIMULASI





TERIMAKASIH

- KELOMPOK 4 -