ITMO

image segmentation with data collection from www.flickr.com database

Ma Cheng Yuan
J4133C
372818

Content

ITMO

introduction :

-flickr database and goal of project
-implemented tools

practice :

[1] data fetch [2] VGG image annotator [3] Pyspark

[4] cv2 [5] split datasets [6] training model

[7] loss [8] Result

flickr database

url : www.flickr.com

description : Flickr is an American image hosting and video hosting service, as well as an online community,

method : using flickr api

desired datasets : blue jay, northern cardinal, american goldfinch , each 20 images

purpose : image segmentation

tools

- PyTorch
- FlickrAPI
- Pyspark
- CV2
- Albumentations
- VGG image annotator

[1] data fetch

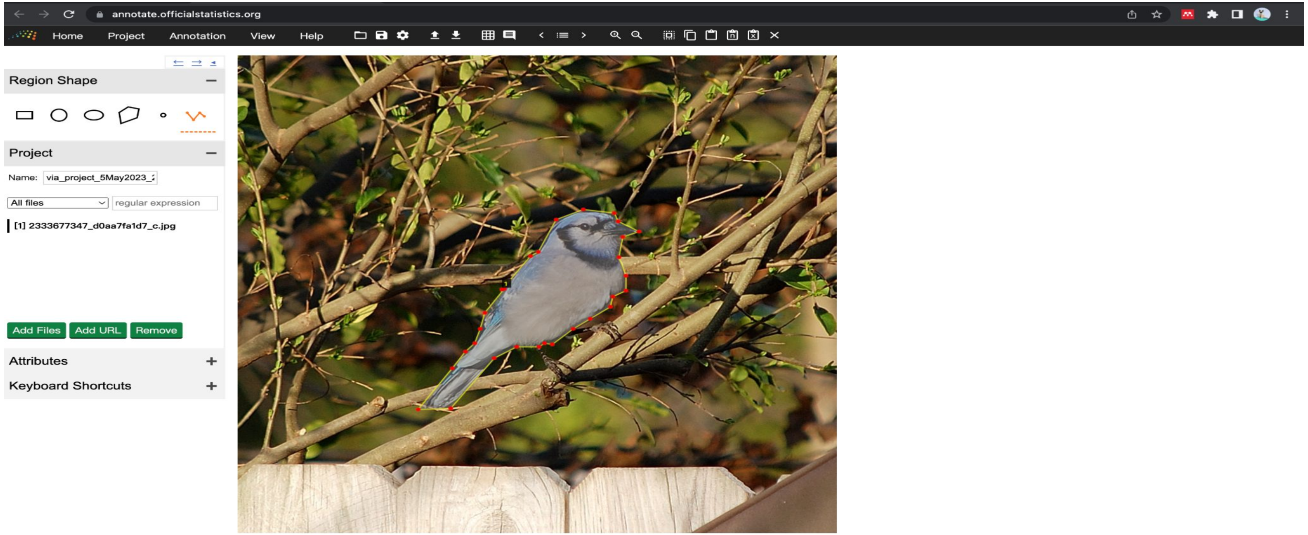
ITMO

by using flickr api , it is easy to download desired image with desired requirement , such as

- image size
- media type(image,video)
- image type (screenshot , photo)
- location (indoor , outdoor)
- image per page
- page to search

[2] VGG image annotator

creation of mask images by using vgg image annotator



Name: via_project_5May2023_

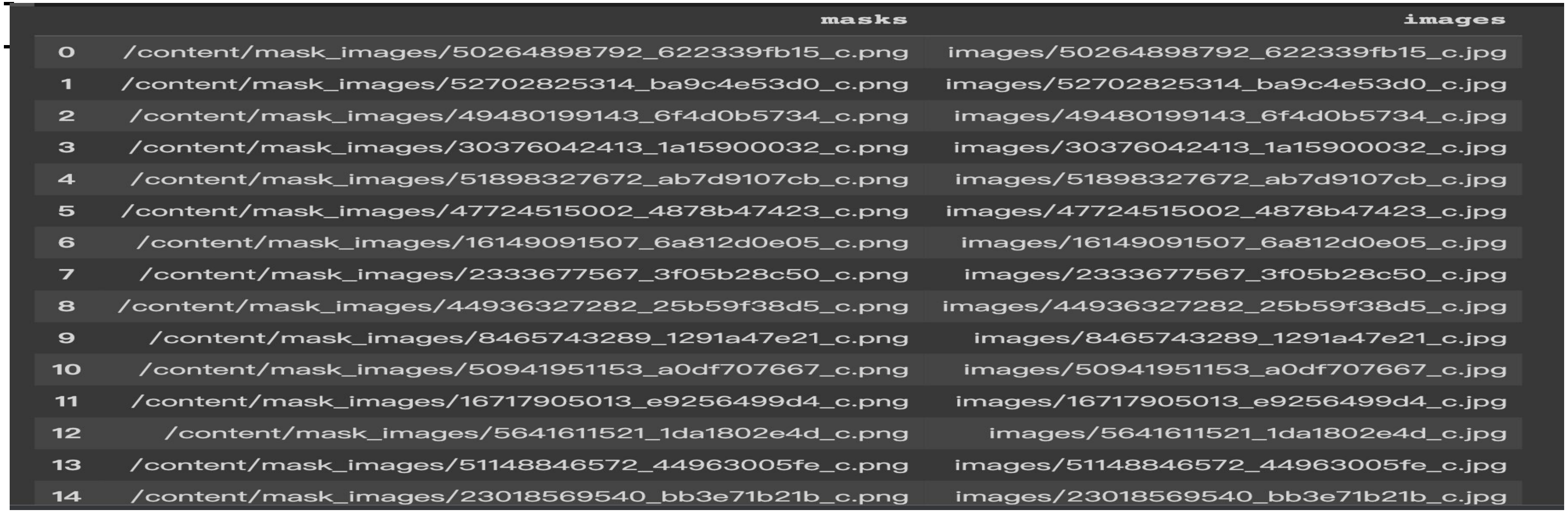
All files

regular expression

[1] 2333677347_d0aa7fa1d7_c.jpg

[3] Pyspark

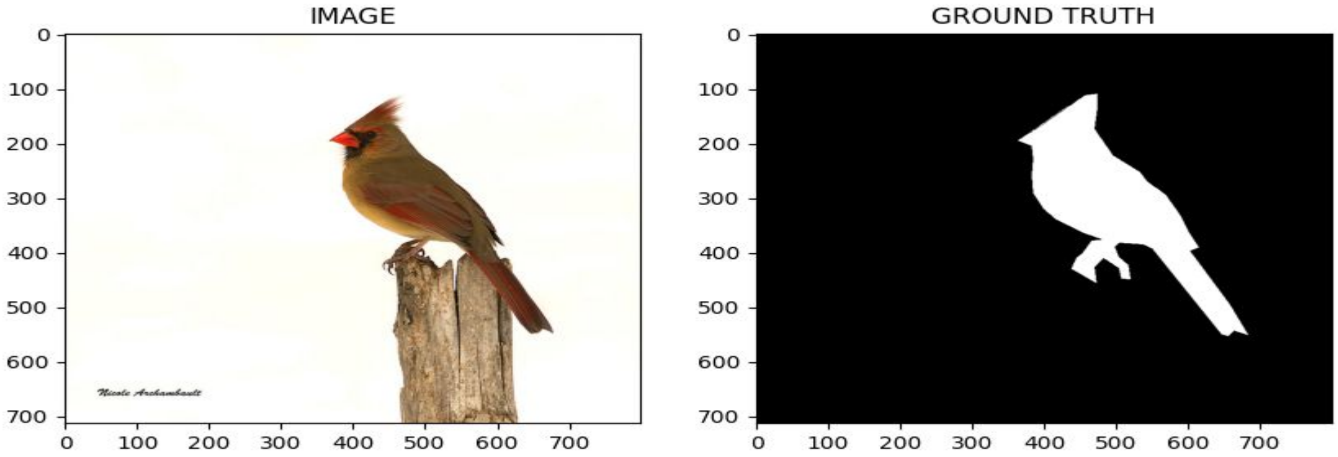
- after creation of images with border of object , we upload them and combine with real images to make a datasets by using PySpark interface to access Pandas API through Apache Spark framework



	masks	images
0	/content/mask_images/50264898792_622339fb15_c.png	images/50264898792_622339fb15_c.jpg
1	/content/mask_images/52702825314_ba9c4e53d0_c.png	images/52702825314_ba9c4e53d0_c.jpg
2	/content/mask_images/49480199143_6f4d0b5734_c.png	images/49480199143_6f4d0b5734_c.jpg
3	/content/mask_images/30376042413_1a15900032_c.png	images/30376042413_1a15900032_c.jpg
4	/content/mask_images/51898327672_ab7d9107cb_c.png	images/51898327672_ab7d9107cb_c.jpg
5	/content/mask_images/47724515002_4878b47423_c.png	images/47724515002_4878b47423_c.jpg
6	/content/mask_images/16149091507_6a812d0e05_c.png	images/16149091507_6a812d0e05_c.jpg
7	/content/mask_images/2333677567_3f05b28c50_c.png	images/2333677567_3f05b28c50_c.jpg
8	/content/mask_images/44936327282_25b59f38d5_c.png	images/44936327282_25b59f38d5_c.jpg
9	/content/mask_images/8465743289_1291a47e21_c.png	images/8465743289_1291a47e21_c.jpg
10	/content/mask_images/50941951153_a0df707667_c.png	images/50941951153_a0df707667_c.jpg
11	/content/mask_images/16717905013_e9256499d4_c.png	images/16717905013_e9256499d4_c.jpg
12	/content/mask_images/5641611521_1da1802e4d_c.png	images/5641611521_1da1802e4d_c.jpg
13	/content/mask_images/51148846572_44963005fe_c.png	images/51148846572_44963005fe_c.jpg
14	/content/mask_images/23018569540_bb3e71b21b_c.png	images/23018569540_bb3e71b21b_c.jpg

[4] cv2

- after that using cv2 with images with border of object to make mask images with gray scale



[5] split datasets

- split datasets into training dataset and validation datasets with proportion 5 : 1

- also by using Albumentations we unify the size of image by resizing 320 x 320 in both training datasets and validation datasets

- In training sets , by using Albumentations we implement 50% of possibility to commit horizontal flip and vertical flip to increase more diversities , so model can avoid overfitting and more practical

[6] training model

- we implement API segmentation_models_pytorch to train model , it is a Python library with Neural Networks for Image Segmentation based on PyTorch

- While training a model, by using API DataLoader we pass samples in “minibatches”, reshuffle the data at every epoch to reduce model overfitting, and use Python’s multiprocessing to speed up data retrieval.

- We have packed datasets into batches, each with 3 pairs (image, ,mask image)

[6] Unet

- Unet is a fully convolution neural network for image semantic segmentation

  encoder_name - name of classification model (without last dense layers) used as feature extractor to build segmentation model (efficientnet-b0)

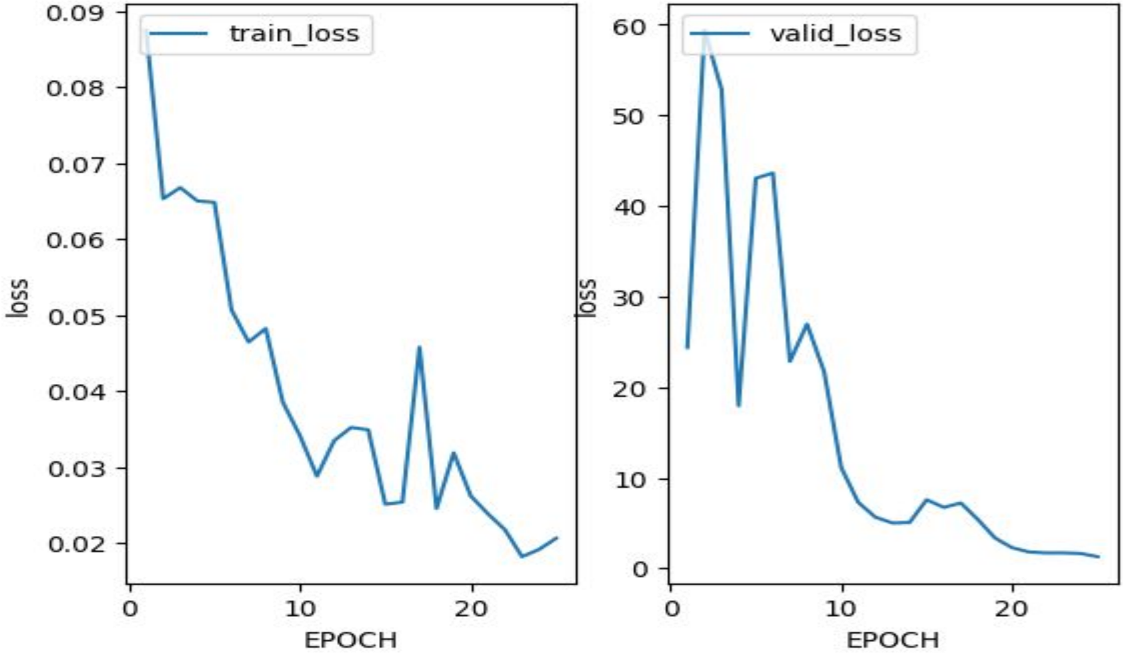
  encoder_weights - imagenet (pre-training on ImageNet)

  in_channels - number of input channels for model, default is 3

  classes - a number of classes for output (output shape - (batch, classes, h, w))

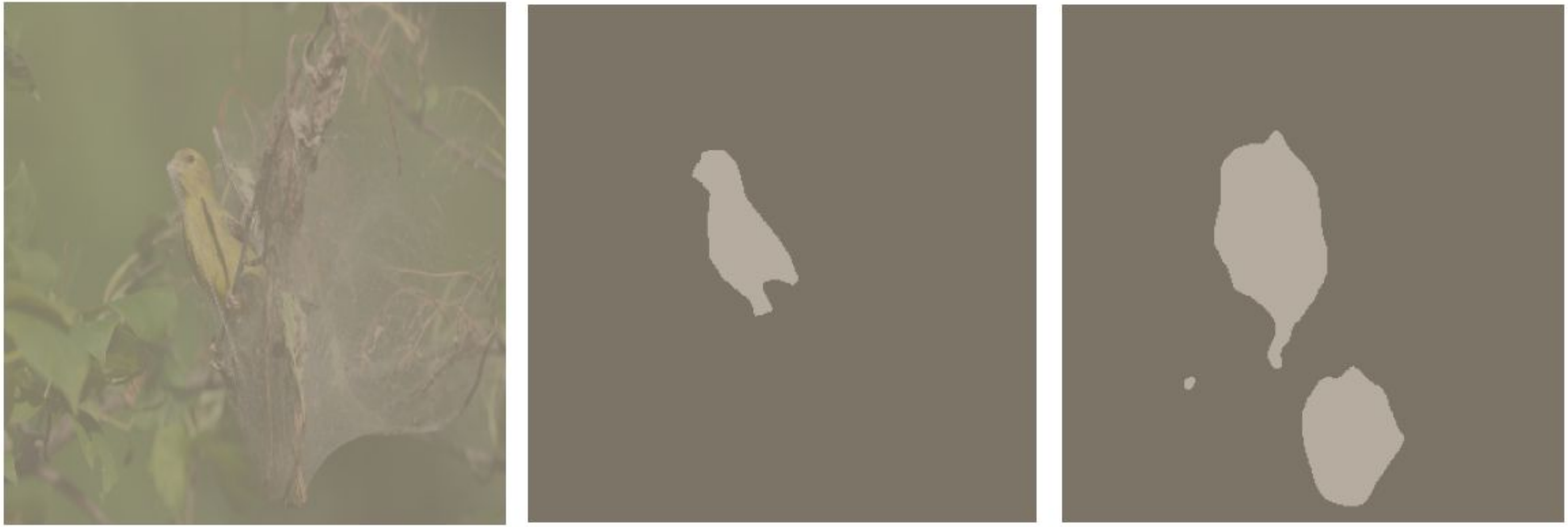
  activation - activation function to apply after final convolutionxa

[7] loss

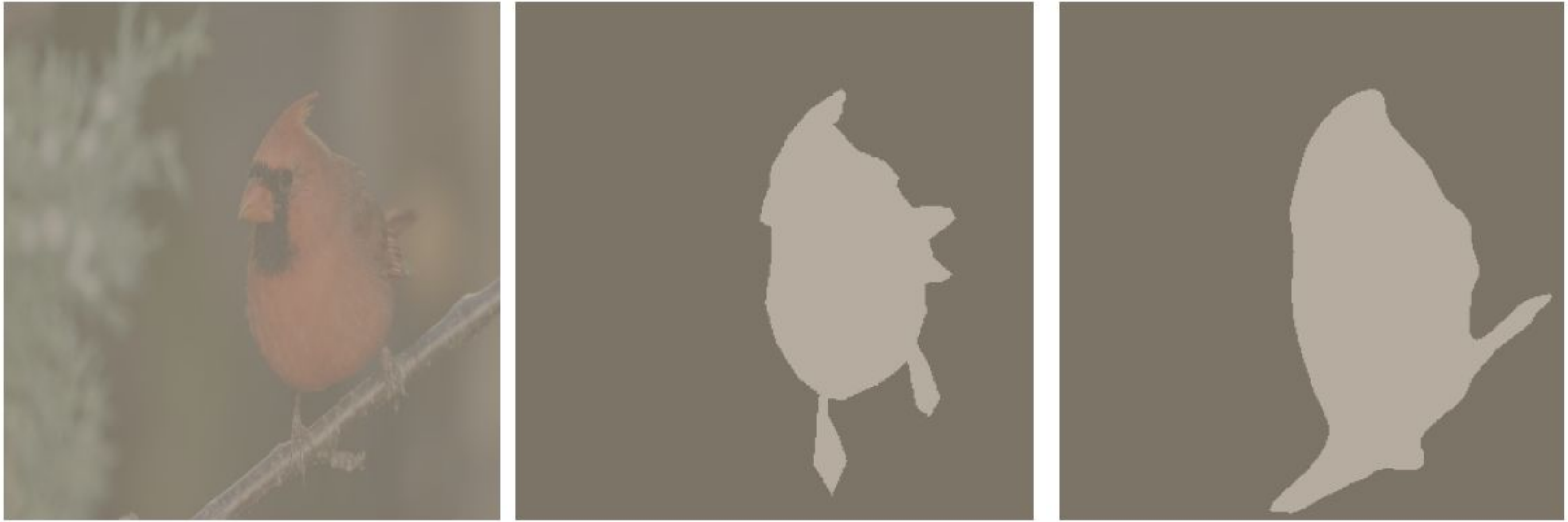


train_loss
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
loss
0
10
20
EPOCH
valid_loss
60
50
40
30
20
10
0
loss
0
10
20
EPOCH

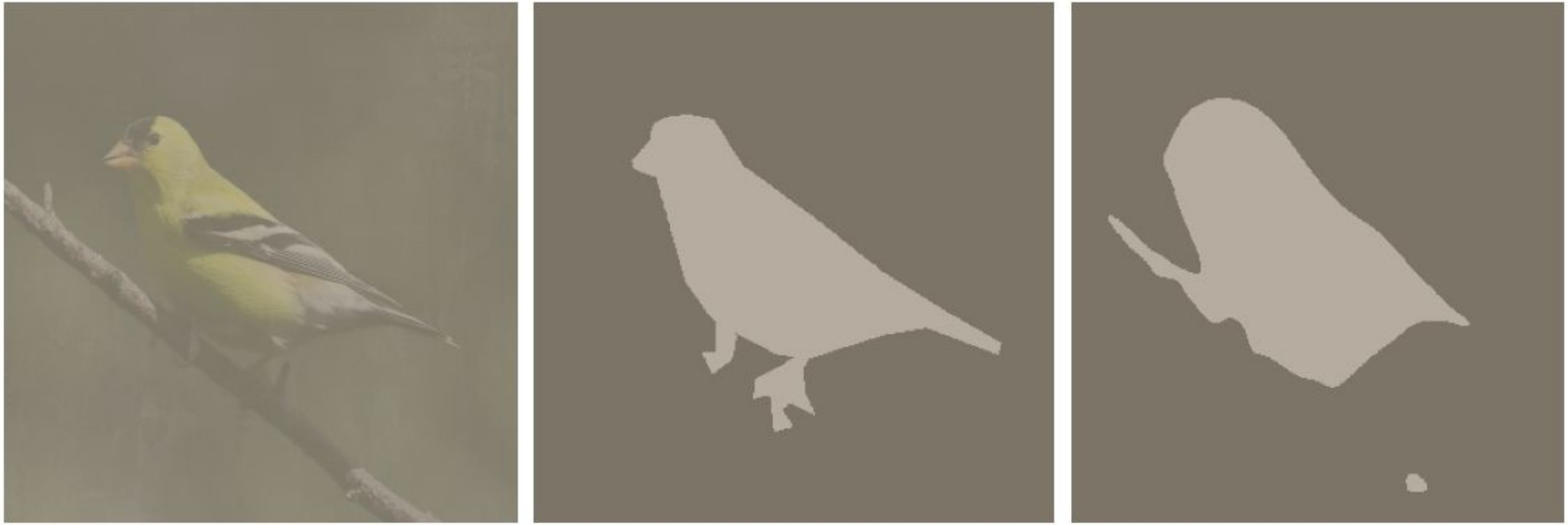
[8] Result



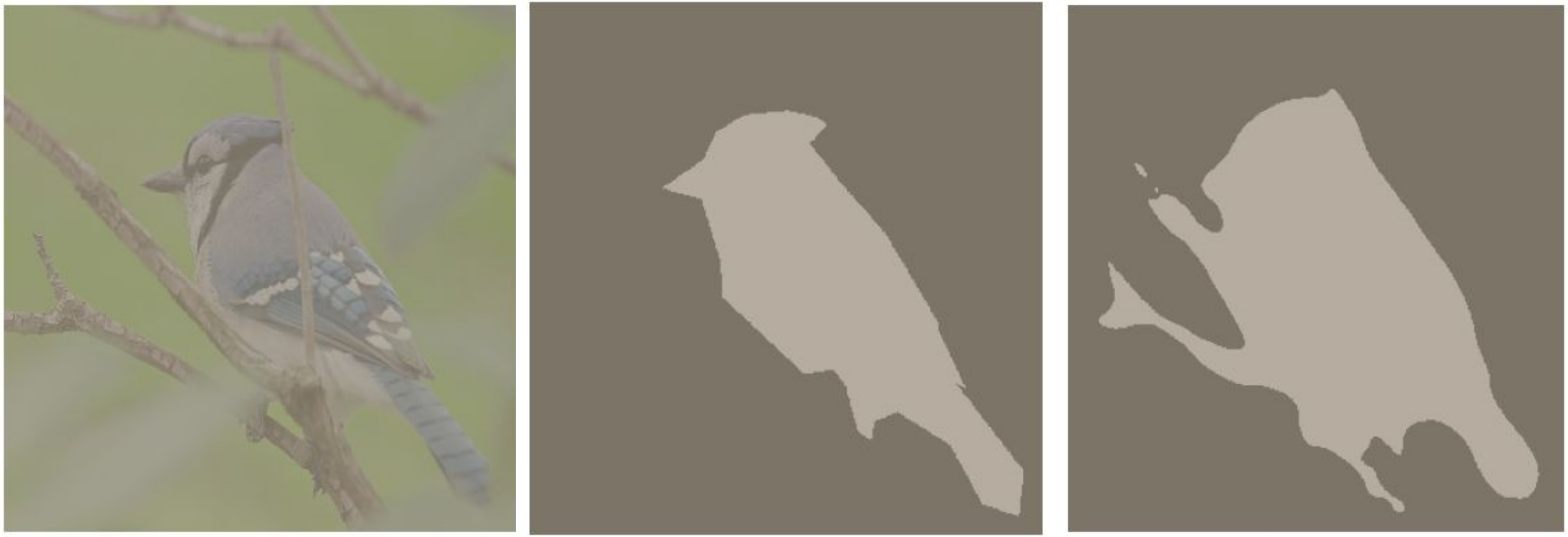
[8] Result



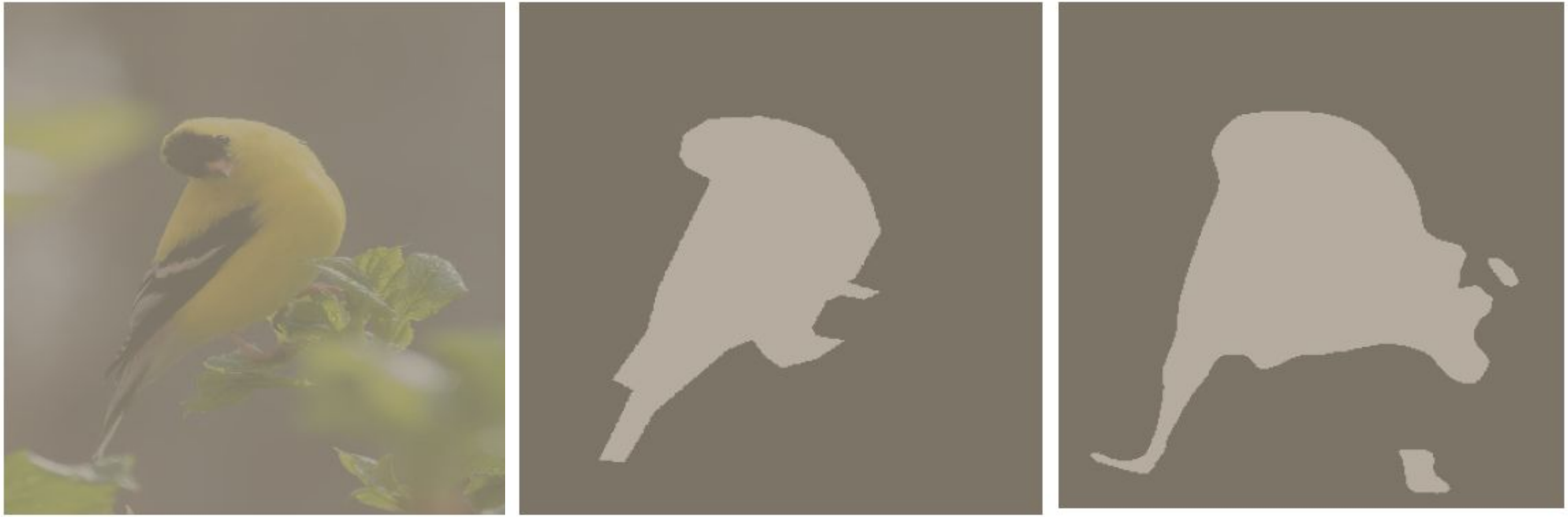
[8] Result



[8] Result



[8] Result



THANK YOU
FOR YOUR TIME!

ITsMOre than a UNIVERSITY