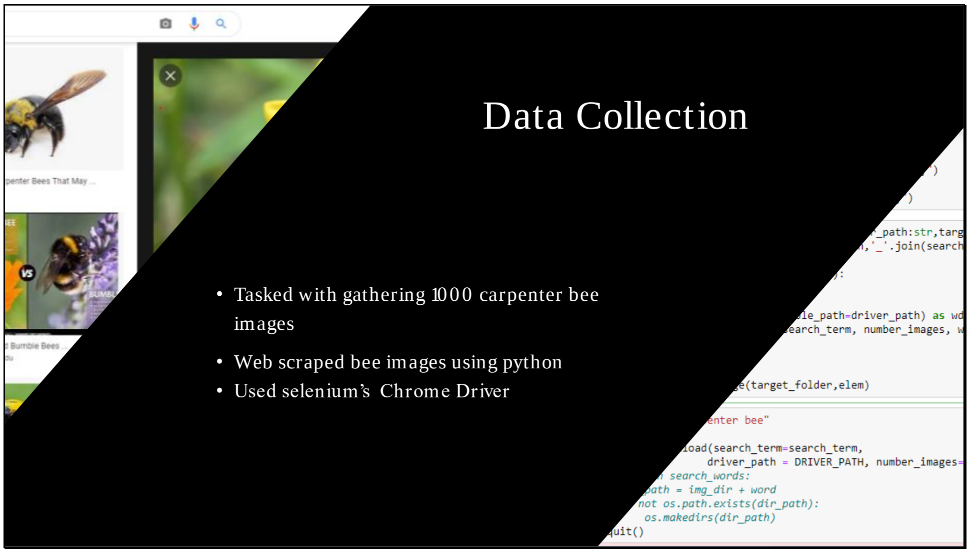
Slide 1

BEES PROJECT

Logan Morrill

Slide 2



Data Collection

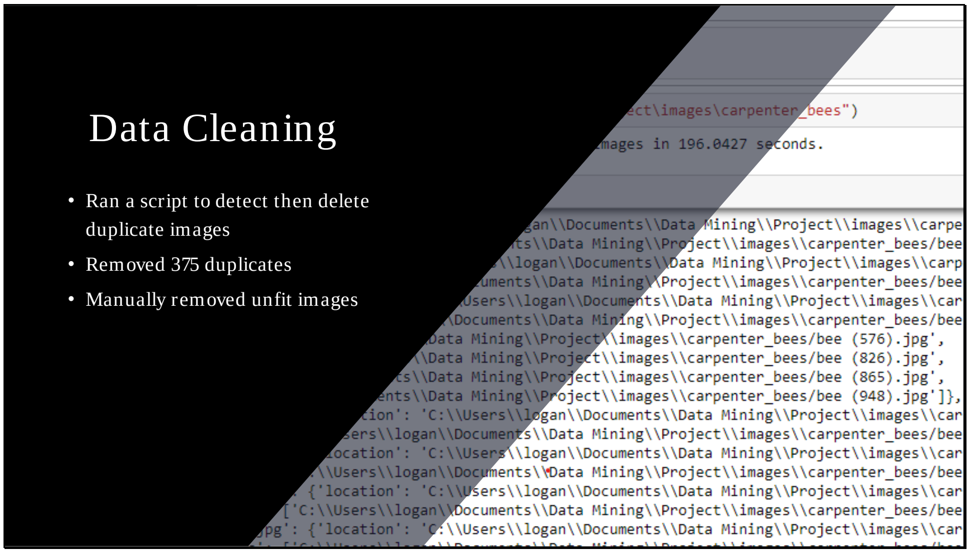
- Tasked with gathering 1000 carpenter bee images
- Web scraped bee images using python
- Used selenium’s Chrome Driver

```
r_path:str,targ
n,'_'.join(search
):

ble_path=driver_path) as wd
search_term, number_images, w

ge(target_folder,elem)

enter bee"

load(search_term=search_term,
          driver_path = DRIVER_PATH, number_images=
n search_words:
path = img_dir + word
not os.path.exists(dir_path):
 os.makedirs(dir_path)
quit()
```

Gather about 1500 because I would have to remove duplicates
The python script searched Google and automatically downloaded image then
Went to download the next image.
Saved lots of time over download 100 images manually

Was the most interesting using python to automatically download images.

Slide 3



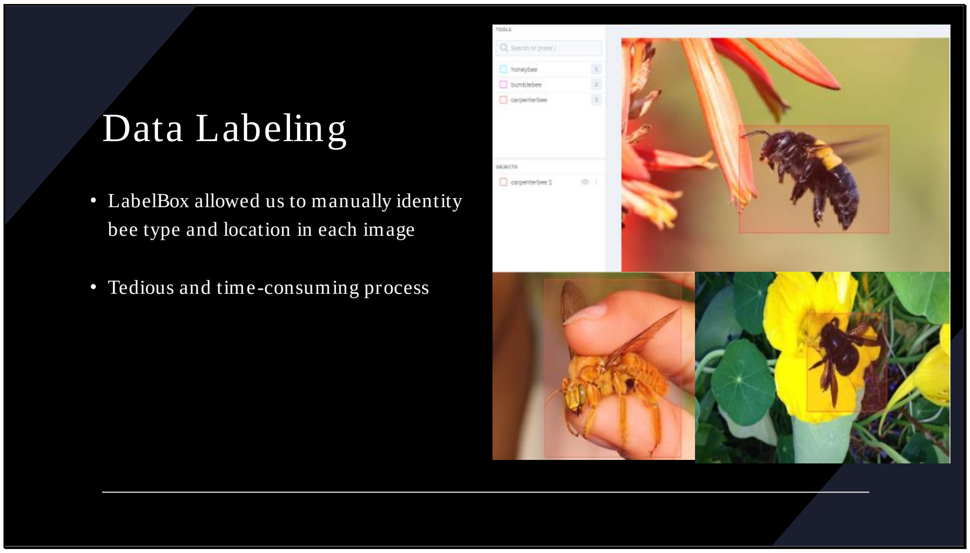
Data Cleaning

- Ran a script to detect then delete duplicate images
- Removed 375 duplicates
- Manually removed unfit images

```
ect\images\carpenter_bees")
images in 196.0427 seconds.

gan\\Documents\\Data Mining\\Project\\images\\carpe
ts\\Data Mining\\Project\\images\\carpenter_bees/bee
\\logan\\Documents\\Data Mining\\Project\\images\\carp
uments\\Data Mining\\Project\\images\\carpenter_bees/bee
Users\\logan\\Documents\\Data Mining\\Project\\images\\car
\Documents\\Data Mining\\Project\\images\\carpenter_bees/bee
Data Mining\\Project\\images\\carpenter_bees/bee (576).jpg',
\\Data Mining\\Project\\images\\carpenter_bees/bee (826).jpg',
ts\\Data Mining\\Project\\images\\carpenter_bees/bee (865).jpg',
ents\\Data Mining\\Project\\images\\carpenter_bees/bee (948).jpg']},
tion': 'C:\\Users\\logan\\Documents\\Data Mining\\Project\\images\\car
sers\\logan\\Documents\\Data Mining\\Project\\images\\carpenter_bees/bee
location': 'C:\\Users\\logan\\Documents\\Data Mining\\Project\\images\\car
\\Users\\logan\\Documents\\Data Mining\\Project\\images\\carpenter_bees/bee
{'location': 'C:\\Users\\logan\\Documents\\Data Mining\\Project\\images\\car
['C:\\Users\\logan\\Documents\\Data Mining\\Project\\images\\carpenter_bees/bee
jpg': {'location': 'C:\\Users\\logan\\Documents\\Data Mining\\Project\\images\\car
```

Throughout the process of web scraping, the script downloaded duplicate images.
I ran another script to detect then remove duplicates. It removed 375 duplicates.
After, went in and manually removed images that I thought were unfit. Such as cartoons, images that didn't contain a bee, or unclear images.

Slide 4

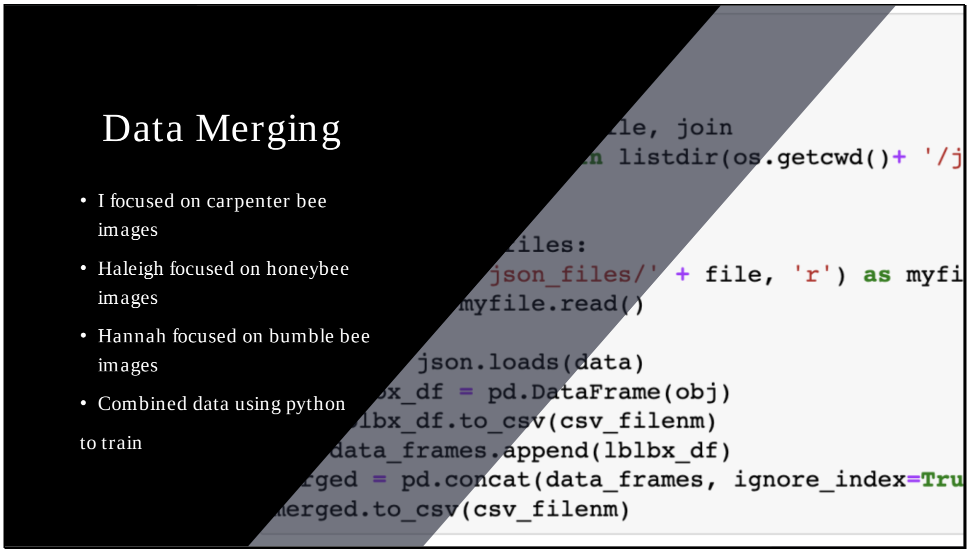


Data Labeling

- LabelBox allowed us to manually identity bee type and location in each image
- Tedious and time-consuming process

Labeled the data using labelbox.
Had to manually look at each of the image and find and label the bees in each of the 1000 images.
Took the most time identifying each bee in each image. Took about 7 hours.

Slide 5



Data Merging

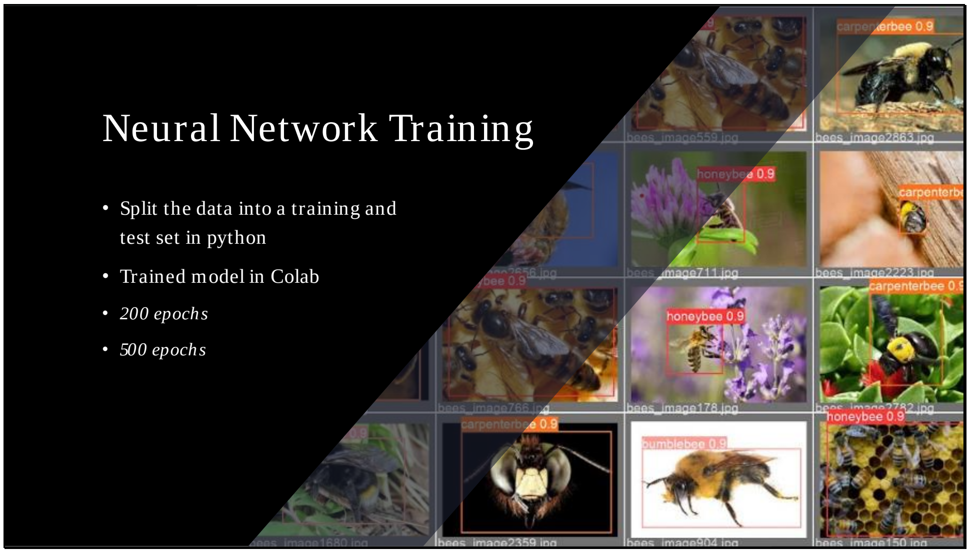
- I focused on carpenter bee images
- Haleigh focused on honeybee images
- Hannah focused on bumble bee images
- Combined data using python

to train

```
               le, join
              n listdir(os.getcwd()+ '/j


           iles:
          json_files/' + file, 'r') as myfi
         myfile.read()

        json.loads(data)
       x_df = pd.DataFrame(obj)
      lbx_df.to_csv(csv_filenm)
      data_frames.append(lblbx_df)
    rged = pd.concat(data_frames, ignore_index=Tru
    erged.to_csv(csv_filenm)
```

Each of us were in charge of gather, cleaning, and labeling one type of bee. Once we were done we combined the data to begin to make our model.

Slide 6

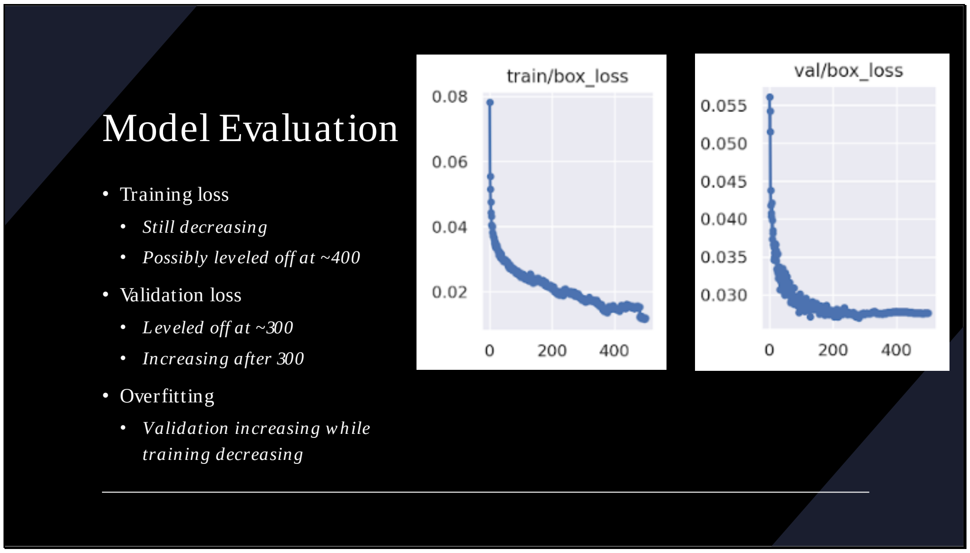


Neural Network Training

- Split the data into a training and test set in python
- Trained model in Colab
- *200 epochs*
- *500 epochs*

We split the data into 80% training data and 20% test data.
We trained the model using Google Colab.
Initially trained 200 epochs, but it wasn’t enough so I bumped it up to 500.
I started training a model with 600 epoch with large weights, but I decided to scrap it because it was taking too long.
Most challenging getting boxes from labelbox and converting them to a way colab could read them

Slide 7

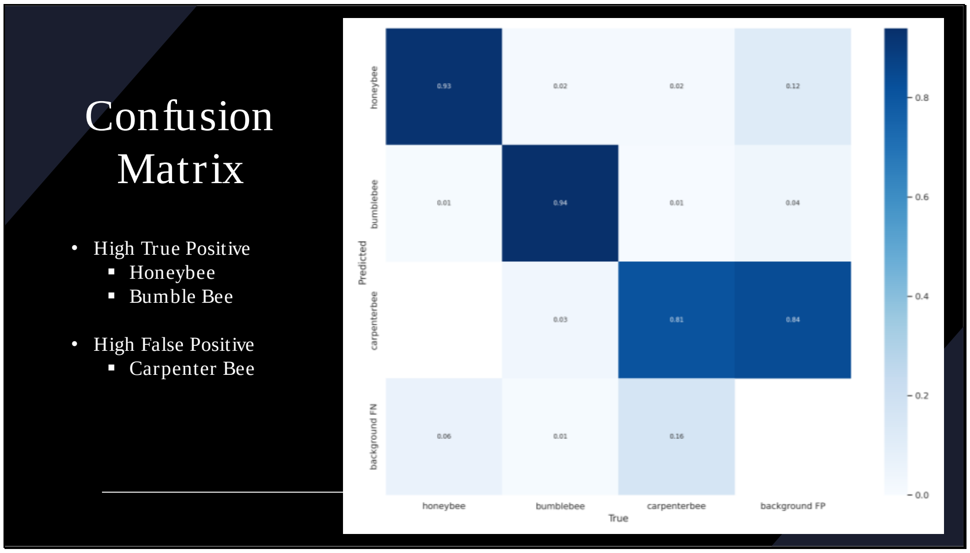


Model Evaluation

- Training loss
  - *Still decreasing*
  - *Possibly leveled off at ~400*
- Validation loss
  - *Leveled off at ~300*
  - *Increasing after 300*
- Overfitting
  - *Validation increasing while training decreasing*

Looking at the losses of the model.
The training loss has is decreasing meaning the model is improving.
Starts to kinda level off at about 400 epochs.
The validation loss decreases rapidly, it levels off at around 300 epochs, then starts to increase.
If the training loss is decreasing or flatting out, while the validation is increasing.
This means the model is overfitting to the training data.

Slide 8



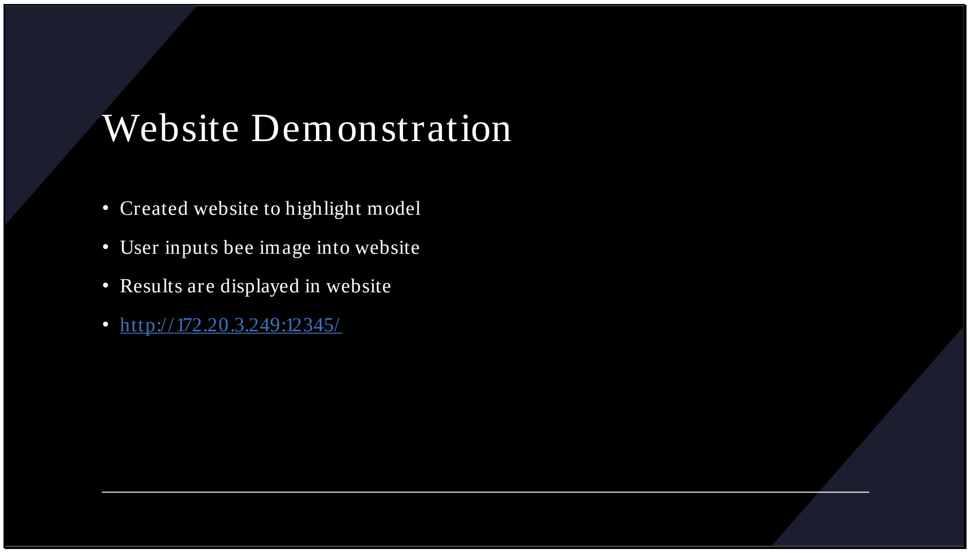
Confusion Matrix

- High True Positive
  - Honeybee
  - Bumble Bee
- High False Positive
  - Carpenter Bee

	honeybee	bumblebee	carpenterbee	background FP
honeybee	0.93	0.02	0.02	0.12
bumblebee	0.01	0.94	0.01	0.04
carpenterbee		0.03	0.81	0.84
background FN	0.06	0.01	0.16	

The confusion matrix tells us some interesting things.
Honeybee and bumblebee high true positive.
Meaning the model predicted the bee type and it was the correct bee 90% of the time.
Carpenter Bee had a high false positive.
Meaning the model predicted carpenter bee but was actually just the background.
Since I was in charge of gather carpenter bee data I could tell you why this happened.
The data a gathered had lots of different types of carpenter bees.
Some were yellow, some were blue, some were purple.
With multiple different colors this made it hard for model to distinguish between bee and background.
One thing that was interesting was the false negative for carpenter bees was not real high.
Meaning the model didn’t predict background to be a carpenterbee.
This tells us the model was more likely to say it detected a bee when it did not.

Slide 9



Website Demonstration

- Created website to highlight model
- User inputs bee image into website
- Results are displayed in website
- http://172.20.3.249:12345/

Once we had finished the model we created a website with the model built in.
A user can upload an image of a bee into the website and the website will spit out a result.
Results are displayed in the website
Lets take a look at the website.
Go over website.
Section that goes into detail about the neural network model.
Section with a description of each bee type.
Footer with everyone in the class.