[Quick links](#_dr9mr0qyx0m2)

[Working Presentation](#_ai05aeh8f74u)

[Github](#_vsmoly4q81xf)

[Useful Commands](#_cnupuqt72gvr)

[To Mount Google Cloud Storage As a Shared Folder](#_hdzb9i6tx1du)

[A Simple Sketch Program](#_8qfwcmua7kyd)

[Downloading stuff from the bucket](#_h7qxnzr9yhs5)

[Questions](#_e7uyu0aw6e45)

[Week 3](#_g9n3ekjhr8xi)

[08/04/2020](#_rx2zz4xom5gk)

[To do](#_dnd52lgrwcf3)

[Results](#_sw11lk5w4zy2)

[08/05/2020](#_60thgnqhd5d4)

[To do:](#_9o9emcwql8q1)

[Results](#_p3s991q8hgi0)

[08/06/2020](#_gxra4ijy4amc)

[To do](#_iukshrtzyyuc)

[Results](#_k2shpxwpu40i)

[Week 4](#_9k2tvki9b4ni)

[To do](#_xnsbahox3ahz)

[Results](#_e2ig6lker73d)

[Week 5](#_ky594sbrlq26)

[accomplished:](#_egrpmjaekxqf)

[Next Steps:](#_r7dv657kv8t4)

[Week 6](#_7own99z0z74m)

[To do](#_by1zl35521sq)

[Results](#_tpaz7z1z0wuu)

[Week 7](#_p7tji1kxl91l)

[Accomplished:](#_o14phlp18izu)

[Next steps](#_xyfms8mo9e2k)

[Week 8](#_w3nplnk9lk5p)

[Must do](#_q9vbca3v24tr)

[Will probably do](#_2d0m9rzdohwx)

[Reach](#_cjp1xf75q3hl)

[Goals for Sept 18](#_ehdmgos6ldfv)

[Week 9](#_28jb7zl6cc7p)

[Progress](#_qrfhdqqi2i51)

[Goals for Sept 24th](#_nahqpwx65t0p)

[Week 10](#_n9376u5gqqix)

[To do:](#_69nzoavexdim)

[Reach](#_edodh4up3zy)

# Quick links

## Working Presentation

<https://docs.google.com/presentation/d/1ylm6SUykjBptTaFI6zUyABlDf-R3irVHpqJIB6DxTVQ/edit?usp=sharing>

## Github

<https://github.com/amazurek1/Etsy-sov>

## Useful Commands

for entering docker: docker run -it --runtime=nvidia --rm my\_docker\_name bash

for entering vm: gcloud compute ssh **--**project etsy**-**esv**-**sandbox **--**zone us**-**central1**-**c smazurek-vm

Might have to first do ~/.bash\_profile

to copy files to vm: open the vm in one window and open regular terminal in the other. Then in regular terminal: gcloud compute scp --project etsy-esv-sandbox --zone us-central1-c --recurse FILE\_NAME smazurek-vm:~/

To copy files into docker: go to vm but not in docker in vm

Type: docker ps

Then look at the name and do:

docker cp SBIR\_regression NAME:/opt

-opt is the place you want to put the file into docker

To run file in docker:

Python file\_name

To move file from docker to vm. Go to vm and type in:

docker cp admiring\_golick:/opt/SBIR\_regression/feat.npy feat.npy

To move file from vm to local drive. Go to local and type:

gcloud compute scp --project etsy-esv-sandbox --zone us-central1-c --recurse smazurek-vm:~/SBIR\_regression/feat.npy feat.npy

To run Ben’s UI:

Open anaconda. Click environments. Click play on base (root). Then a few options will come up and click “open terminal”. Then in the terminal cd into the github (Etsy-sov) and then cd etsyOutlineDemo. And then type in: python gridTk.py

**Running Tu’s Model - Steps**

1) Copy files to vm

gcloud compute scp --project etsy-esv-sandbox --zone us-central1-c --recurse ~/SBIR\_regression rjohnsonlaird-vm:~/

2) Login to vm

gcloud compute ssh --project etsy-esv-sandbox --zone us-central1-c rjohnsonlaird-vm

3) Check the docker processes:

docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

c7ce1b1873f1 my\_docker\_name "bash" 57 minutes ago Up 57 minutes jovial\_goodall

4a) If docker is not running:

4b) cd ~/caffe\_l2norm-master

4c) Run docker

docker run -it --runtime=nvidia --rm my\_docker\_name bash

5) Copy files from vm to docker (docker name is the two words from ps cmd in red, above)

docker cp SBIR\_regression/ jovial\_goodall:/opt

6) Run the ML code

python getfeat\_img.py

Note: If the docker instance dies - you have to re-run docker & re-copy the files

**To run Jupyter notebook on VM & view in local Browser**

1. In vm start jupyter notebook with:

jupyter notebook --no-browser --port=8081

1. On your Etsy mac run:

gcloud compute ssh rjohnsonlaird-vm --zone=us-central1-c --project=etsy-esv-sandbox -- -L 8081:localhost:8081

[Enter your password when prompted]

It logs you in & sets up the SSH tunnel.

1. On your Etsy mac open a tab and paste in the full URL printed in step 1, which would start with: http://localhost:8081/?token=

[Note: the 8081 port here is given as an example, it could be any unused port as long as it’s used consistently in all three steps]

**To run Jupyter notebook inside Docker on VM & view in local Browser**

1. On your Etsy mac run:

gcloud compute ssh rjohnsonlaird-vm --zone=us-central1-c --project=etsy-esv-sandbox -- -L 8081:localhost:8081

[Enter your password when prompted]

It logs you in & sets up the SSH tunnel.

1. cd ~/caffe\_l2norm-master/

docker run -v ~/Etsy-sov/SBIR\_regression:/root/Etsy-sov/SBIR\_regression -it -p "8081:8081" --runtime=nvidia --rm my\_docker\_name bash

Note: also sharing the SBIR folder thus avoiding any file copy ;-)

1. In docker:

pip install jupyter

1. In docker:

cd /root/Etsy-sov/SBIR\_regression

jupyter notebook --NotebookApp.token='' --port=8081 --ip=0.0.0.0 --allow-root --no-browser . &

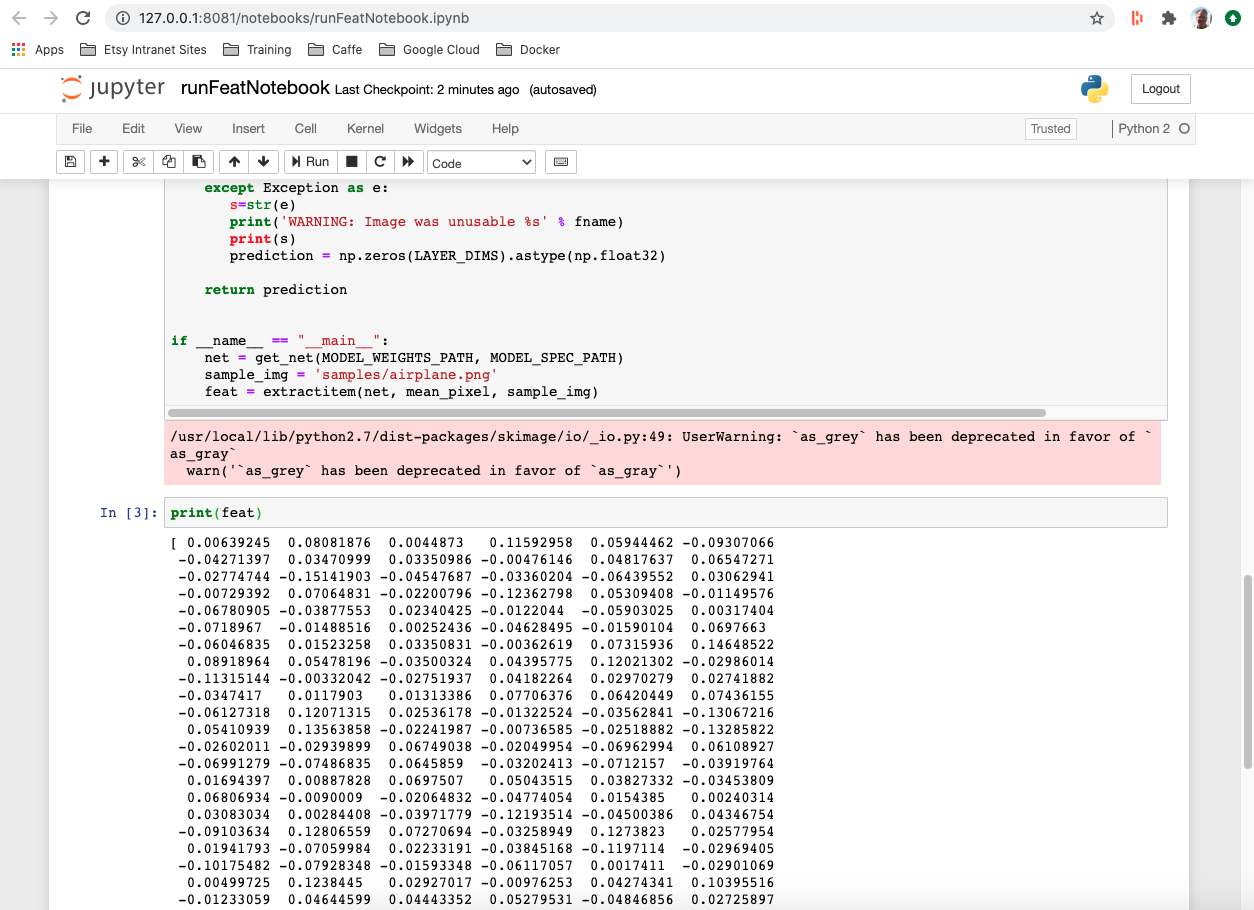
1. On your Etsy mac open url: [http://localhost:8081/](http://localhost:8081/?token=)

Note: the *pip install jupyter* step could be added to the docker build - but then we’d have to rebuild the docker installation from scratch. For now I have put steps 3) & 4) in a shell script here: <https://github.com/amazurek1/Etsy-sov/blob/ben/SBIR_regression/bin/runJupyter.sh> .. If you download the script and put in the SBIR\_regression folder on your vm - you can run inside docker by typing: ~/SBIR\_regression/bin/runJupyter.sh

1. I put a working notebook on my branch here: <https://github.com/amazurek1/Etsy-sov/blob/ben/SBIR_regression/runFeatNotebook.ipynb>

Note I changed the location of the model file to be in the model directory.

If everything works, you should see something like this:



## 

## To Mount Google Cloud Storage As a Shared Folder

1. Login to vm
2. Install homebrew from: <https://brew.sh/>
3. Install gcsfuse:

<https://github.com/GoogleCloudPlatform/gcsfuse/blob/master/docs/installing.md>

brew install gcsfuse

sudo ln -s /usr/local/sbin/mount\_gcsfuse /sbin

1. Make an empty mount point directory

cd

mkdir bucket

1. Append this line to /etc/fstab by editing as sudo [I can help folks do this - as it’s a sensitive change]

etsy-esv-sandbox-storage /home/rjohnsonlaird/bucket gcsfuse rw,user,noauto,implicit\_dirs 0 0

1. mount /home/rjohnsonlaird/bucket
2. cd bucket/SBIR\_images

[Note: you cannot list the contents of the bucket directly for some reason]

**Other info**

Basic Usage

<https://github.com/GoogleCloudPlatform/gcsfuse/blob/master/docs/mounting.md>

# Unmount command

sudo fusermount -u ~/bucket

## A Simple Sketch Program

1. Run brew install tcl-tk
2. Run the notebook sketchPaint.ipynb
3. TODO - add save png functionality

## Downloading stuff from the bucket

Log into VM and then type this in:

gsutil cp -r gs://etsy-esv-sandbox-storage/images/lampshades/il /home/smazurek/Etsy-sov/SBIR\_regression/lampshades\_images

## Questions

Why do we have to so ~/ all the time on the VM

GSUTIL in our commands

# Week 3

## 08/04/2020

### To do

1. Research image to image neuronal networks by tomorrow 2 PM
2. Review with group and try to get a code draft out
3. If it seems promising we will continue, if not look into others

### Results

1. Good paper with code for finding similar images from sketches:

<https://arxiv.org/pdf/1703.05605v1.pdf>

1. Hub for good papers about what we are trying to do. Number 3: Sketch based image retrieval is good:

<https://github.com/MarkMoHR/Awesome-Sketch-Based-Applications#3-sketch-based-image-retrieval-sbir>

1. Our final product will be an input of a stick figure of a lamp and will output similar listings to the input. It might be useful on the backend to turn the etsy images into more stick like figures (especially without color so that the similarities the machine picks up on are not color). We wrote code that takes images and makes them into black and white sketches. Its uploaded to github as “image\_to\_pencil\_sketch”

## 08/05/2020

### To do:

1. Start trying out the code for this paper:

<https://arxiv.org/pdf/1703.05605v1.pdf> But do more research because this paper is in Caffe

1. Create powerpoint of proposal

### Results

Good papers/code for “sketch based image retrieval” (SBIR) using python.

1. <https://www.sciencedirect.com/science/article/abs/pii/S0097849317302194?via%3Dihub>

^ paper

<https://github.com/TuBui/SBIR_regression>

^code for it

1. <https://openaccess.thecvf.com/content_ECCV_2018/papers/Sasikiran_Yelamarthi_A_Zero-Shot_Framework_ECCV_2018_paper.pdf>

^paper

1. <https://github.com/ShivaKrishnaM/ZS-SBIR>

^code for it check github branch “chris”

<https://arxiv.org/pdf/1807.11724.pdf>

1. <https://nextjournal.com/nextjournal/sketch-rnn-magenta>

^code along article

Google pressy for proposal sharable link

<https://docs.google.com/presentation/d/1ylm6SUykjBptTaFI6zUyABlDf-R3irVHpqJIB6DxTVQ/edit?usp=sharing>

## 08/06/2020

### To do

1. Start trying to replicate code from papers from yesterday
2. Work on Proposal presentation

### Results

# Week 4

### To do

1. Building test zero-shot model using VGG pretrained on imagenet
   1. Using labeled and unlabeled data will increase numbers!
2. After we get working model, configure it or find a similar pretrained for our purposes
3. Start to organize our group’s dataset (we will need alot)
4. Evaluate results

### Results

Met with Tu Bui about paper #1 from 08/05

(<https://www.sciencedirect.com/science/article/abs/pii/S0097849317302194?via%3Dihub>

^ paper

<https://github.com/TuBui/SBIR_regression>

^code for it)

Set up docker on local drive, and set up VM and docker on on VM

Ran some of Tu’s programs

Immediate next steps:

1. Understand output vectors (need 256 data points for each image - 256 vectors per image)
2. Run all 25k desk lamp images form etsy on the model and save the vectors
3. Run one lamp sketch and get nearest matches from saved vector lamp images

**Longer term - Enhancing the Trained Model**

1. Run baseline existing model. Cluster the output. Look at the X number distinct clusters
2. Use the X clusters in step 1, to inform the category labels
3. Label 1,000 lamp images into X categories informed by step 2. 100 validation & 100 hold out set.
4. Retrain the model
5. The test label for a lamp is defined as the NN centroid lamp category
6. Validate hold-out set & then measure accuracy
7. Iterate on steps 4 thru 6

# Week 5

## accomplished:

1. Ran baseline model with NN. We now have a pipeline where we can input a sketch/image and program will output most similar images from database using basic scikitlearn NN.
2. Wrote scripts to let us run Jupyter notebooks inside Docker on a VM & view in our local browsers.

## Next Steps:

1. We currently only have 115 table lamp images for our hub on the VM, we need to upload all 42K images
2. We need to come up with a lot of lamp sketches
3. Evaluation of model - put together a list of validation methods
   1. Machine evaluation
   2. Experimental human evaluation
4. Get a lot of sketches from google to run program on

# [Week 6](#_7own99z0z74m)

## To do

1. Look at the 115 images we have on the VM
2. Run a bunch of sketches
3. Work on presentation
4. Email Tu to meet up

## Results

1. Wrote an initial freehand sketch program to draw lamp sketches. It’s a python program using matplotlib. Downloadable here: <https://github.com/amazurek1/Etsy-sov/blob/ben/lampSketchBest.py>

To run it, type in a terminal window: python lampSketchBest.py

To draw a line: mouse click and release where you want to start the line. To end the line mouse click and release.

1. Imported 2k table lamp images from etsy.com
2. Wrote new program to generate vectors for these images
3. Ran sketches through program, and program subjectively returning correct output
4. Initial dendrogram clustering of images

Next steps:

1. Get all 40K images into our bucket.
2. Cluster them -- get image centroids for each cluster of somewhere between 200 - 400 images.
3. Up train the image model only with lamp data using contrastive learning with v. dissimilar clusters. Improvement measure is how many images in two far apart clusters are misclassified as being in the wrong (i.e. other) cluster. An obvious cluster to use is anglepoise lamps as they are very distinctive.
4. Run different sketches on all of the images.. Will use our matplotlib sketch program for some of this.
5. We are still discussing how the end-user UI should look like. Possibilities include “flip-book” with three interchangeable sections: shade, stem & base; user selecting from prototypical centroid images, or for advanced users freehand sketch. Or some combination of these.

Things were wondering:

1. Program seems really good. Even if we ran the sketches through a directory that had different types of objects and lamps, I wonder if it would still pick up only the lamps, and even more specifically the relevant lamps. This is something to try. Just to test how good Tu’s model is

# Week 7

## Accomplished:

1. Presented at interns meeting
2. Wrote pipeline for clustering lamp images and creating dendrogram
3. Can now output the archetypical lamp in each cluster
4. Downloaded 30k table lamp images and ran program to get all numpy arrays
5. Ran the two table lamp sketches we have with the 40k images - output is much better and matches the sketches really well

## Next steps

1. Cp numpy arrays and images to bucket from avivas vm
2. Use chris’s archetypical lamp images to help direct finding more lamp sketch images online
3. Output like 200 nearest neighbor images per sketch
4. Run the dendrogram and clustering on all 30k images
5. Evaluate model
   1. All five of us look browse all 30k images and then look at the output of the lamp
6. Retrain model for lamps
   1. Output is looking pretty good as is, so interesting to see if retraining would improve it
7. sd

# Week 8

1. Got all the numpy arrays into the bucket
2. Ran clustering and dendrogram for all 30k images
3. Figured out the best way to create the lamp outlines from Chris’s archetypical lamp images (Photoshop has a really good tool for this)

## Must do

1. Create outlines using photoshop from Chris’s archetypical images
2. Run outlines and get output
3. Bare minimum demo:
   1. Ppt slide for each outline (~5) and its output
   2. Ppt slide for 2 sketches and their output

## Will probably do

Create UI/video demo similar to Tu’s

Or

Create UI where we show etsy.com with the outlines in the filtering setting - can make interactive UI for this

## Reach

1. Retrain model
2. Show the children lamp outlines after user picks on a beginning outline for further filtering - should definitely mention this in the possible UI section

## Goals for Sept 18

1. Have all outlines and sketches and their output
2. MVP presentation
3. UI

# Week 9

## Progress

1. Produced all outlines - photoshop ended up being really time consuming and we ended up using Tiho83 from fiverr.com. We got 24 outlines, well probably only end up using ~10
2. Demo progress -
   1. Have a Wix site and actual UI we are working on - wix is easy to create the layout we want

Video of how the first page of the demo should look (link below) - created on wix. Notice the outlines on the left hand sound for the advanced search. Final demo we will click on those outlines and it will output the images that the program outputs

<https://drive.google.com/file/d/1OWqUyGVV_ghLEL2HQgY1Hus7aMVq3-g4/view?usp=sharing>

1. Uptraining - successful running caffe - figuring out how to train it with the lamp images
2. Expanded Clustering - good for data analysis

## Goals for Sept 24th

1. Finish demo!!
2. Finalize presentation
3. Label the mislabeled lamps for 2 lamp types
   * 1. Compare text based search lamps to our outline output
     2. Label the mislabeled lamps in a lamp category

# Week 10

1. Finished UI and Wix - Was a tough battle, but chose to use Ben’s UI in the end
2. Created evaluation method to measure precision of the model
3. Completed final presentation (95% done)

## To do:

1. Come up with new title for presentation
2. Finish evaluation and evaluation slide - create evaluation graph

## Reach

1. Finish retraining model
2. Supervised clustering using lamp outlines