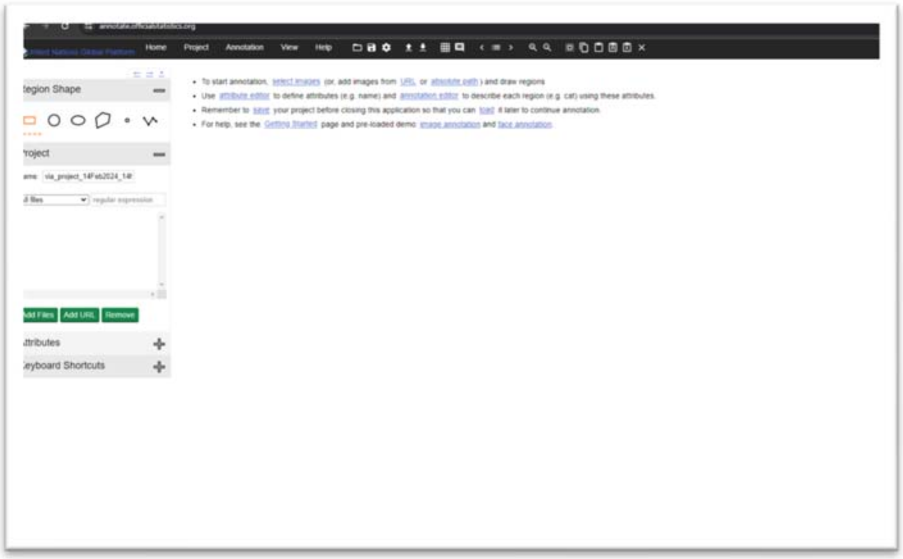
Custom Object Detection GUI Help Guide

Table of Contents

Quick Setup

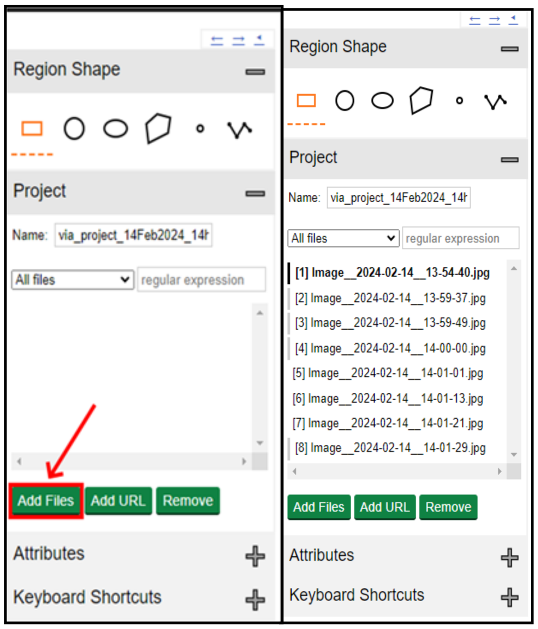
Annotations

1. Gather all images
2. Annotate images using vgg annotation tool. https://annotate.officialstatistics.org/



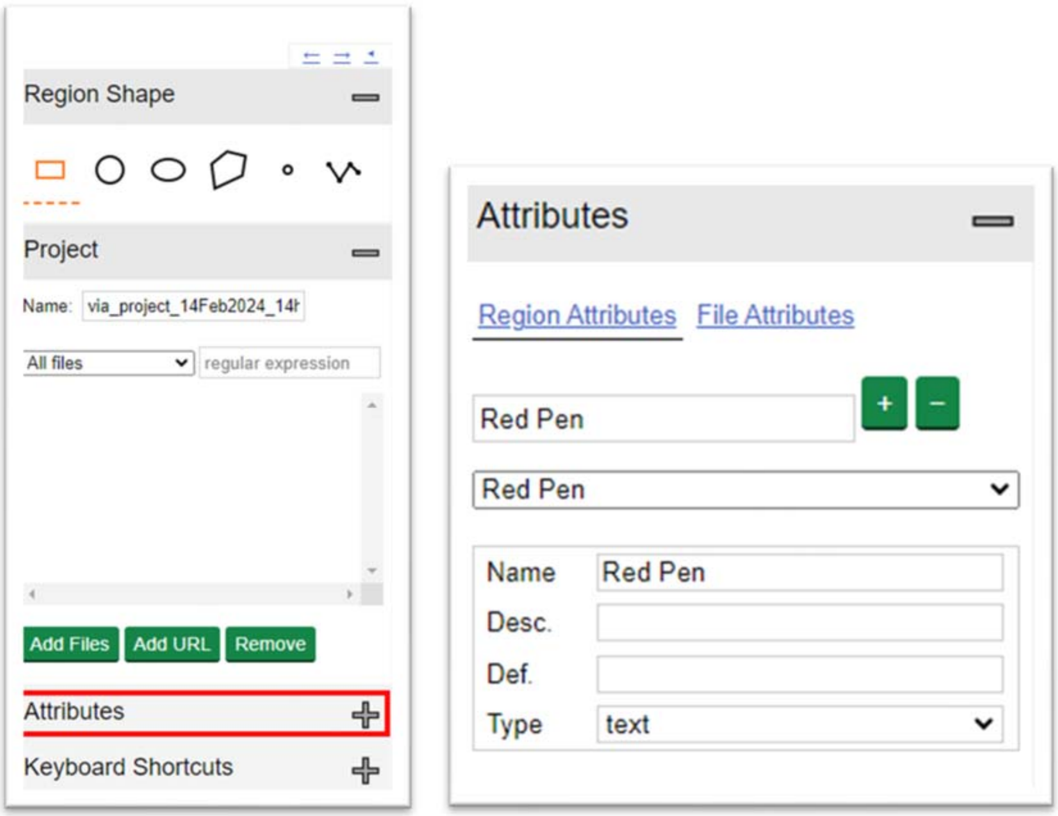
- To start annotation, select images (or, add images from URL or absolute path) and draw regions
- Use attribute editor to define attributes (e.g. name) and annotation editor to describe each region (e.g. cat) using these attributes.
- Remember to save your project before closing this application so that you can load it later to continue annotation.
- For help, see the Getting Started page and pre-loaded demo: image annotation and face annotation.

3. Add your images



[1] Image__2024-02-14__13-54-40.jpg
[2] Image__2024-02-14__13-59-37.jpg
[3] Image__2024-02-14__13-59-49.jpg
[4] Image__2024-02-14__14-00-00.jpg
[5] Image__2024-02-14__14-01-01.jpg
[6] Image__2024-02-14__14-01-13.jpg
[7] Image__2024-02-14__14-01-21.jpg
[8] Image__2024-02-14__14-01-29.jpg

4. Add an attribute (object name).

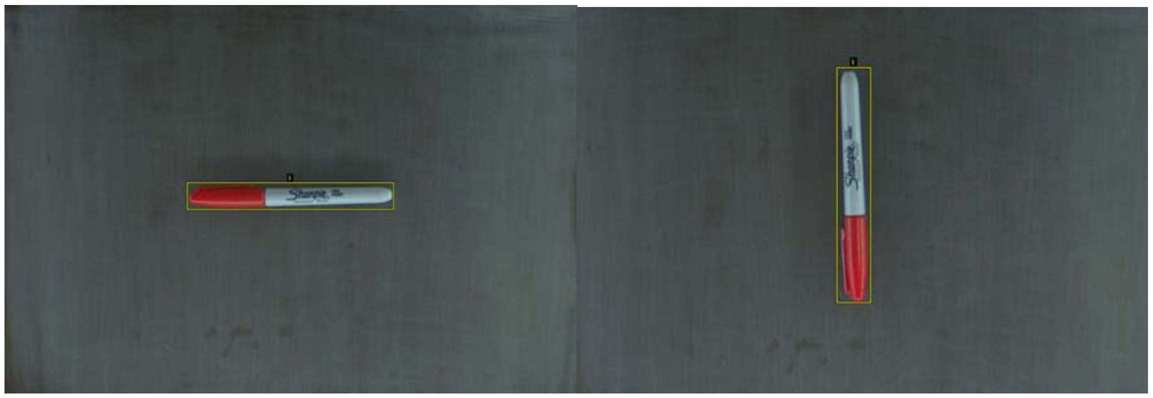


Red Pen

Red Pen

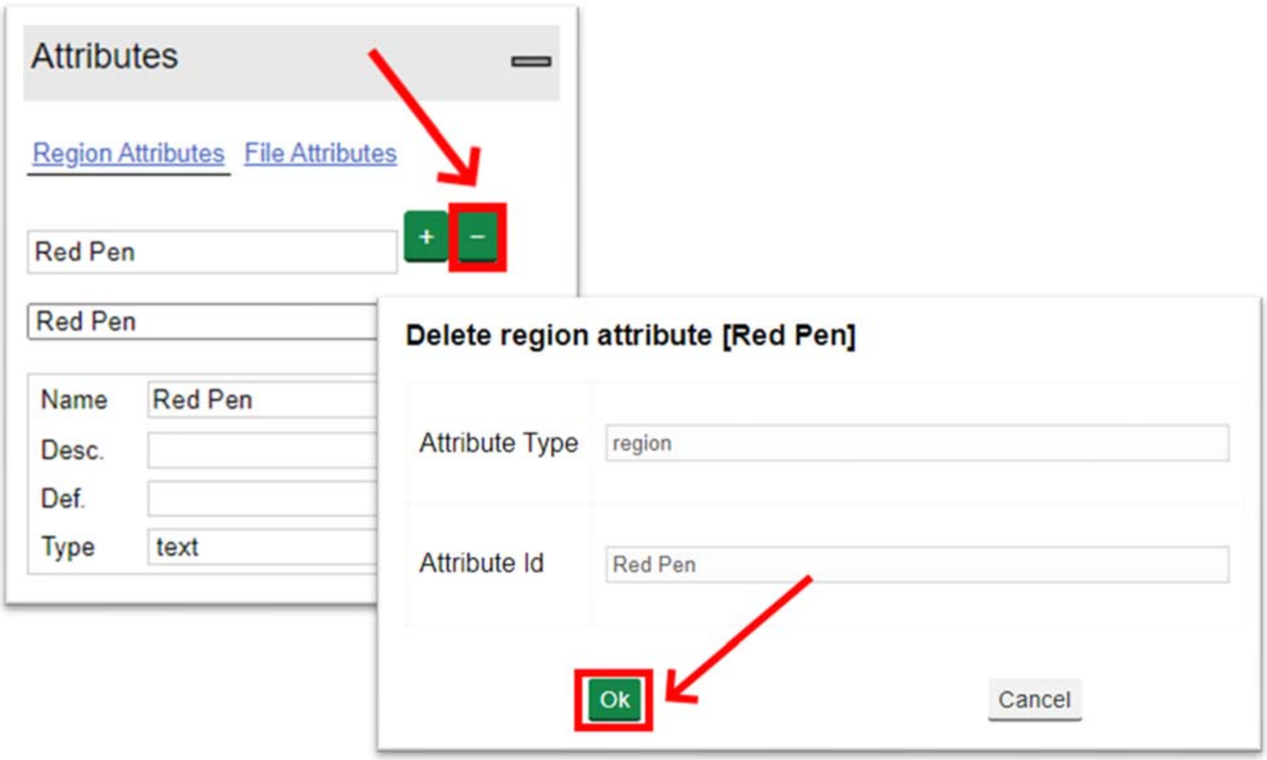
Name	Red Pen
Desc.	
Def.	
Type	text

5. Create bounding boxes around each object of this type in ALL images.



Training with Multiple Object Types

1. After all objects of attribute “X” have been labled, DELETE the attribute and create a new attribute.


Attributes

Region Attributes File Attributes

Red Pen

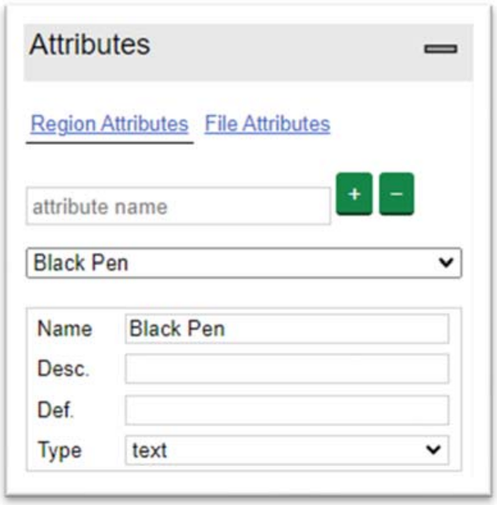
Red Pen

Name	Red Pen
Desc.	
Def.	
Type	text

Delete region attribute [Red Pen]

Attribute Type	region
Attribute Id	Red Pen

2. Add the next object type and begin labeling all objects of that type. Repeat from step 1 for each object type.

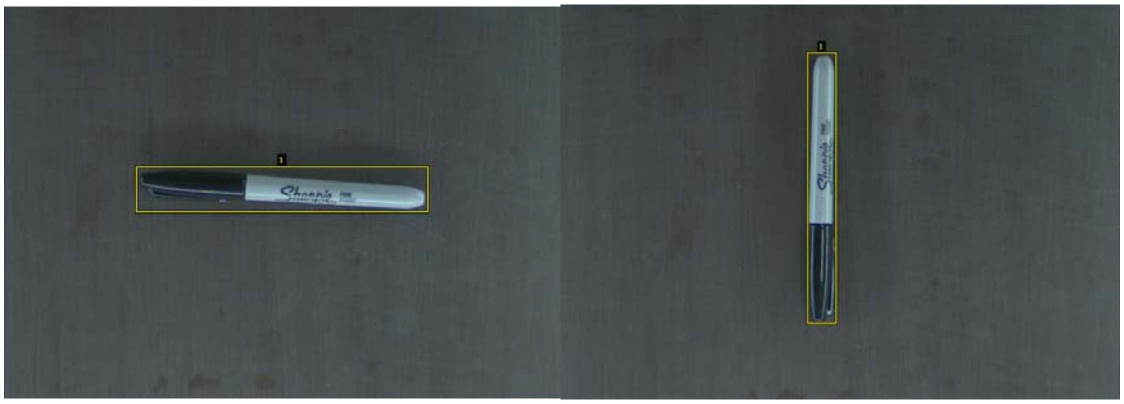

Attributes

Region Attributes File Attributes

attribute name

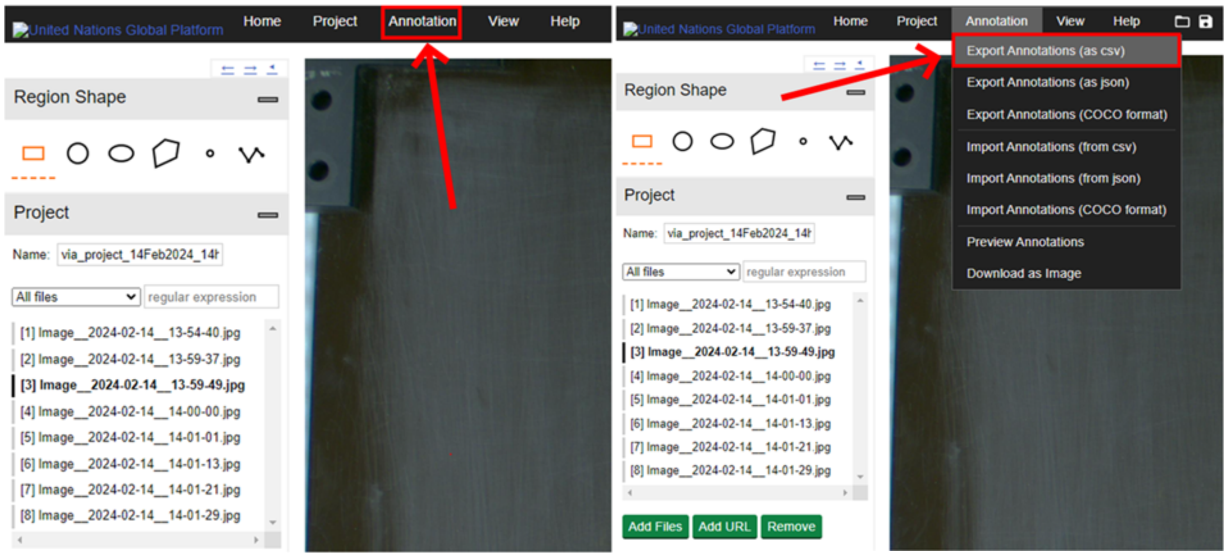
Black Pen

Name	Black Pen
Desc.	
Def.	
Type	text



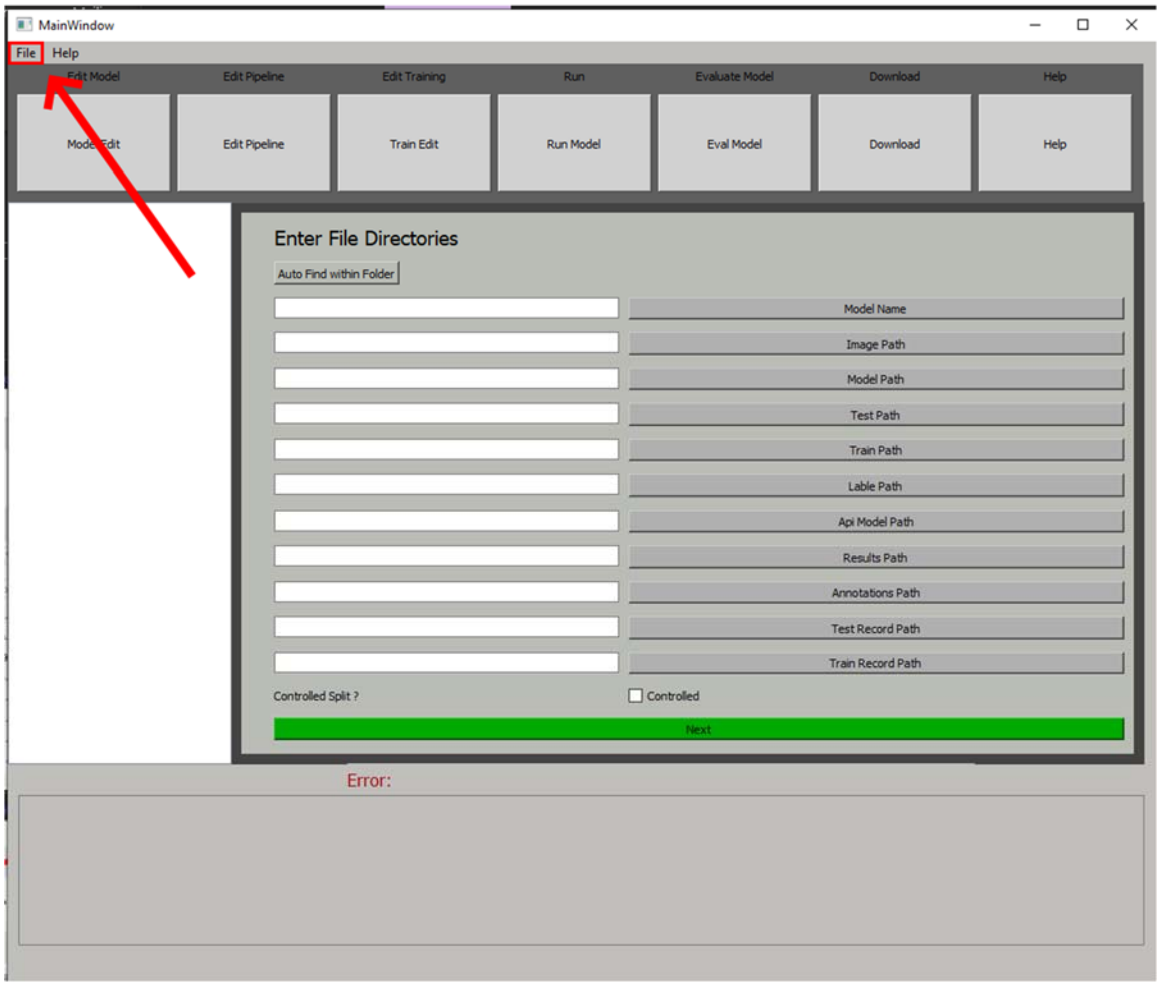
Exporting Annotations

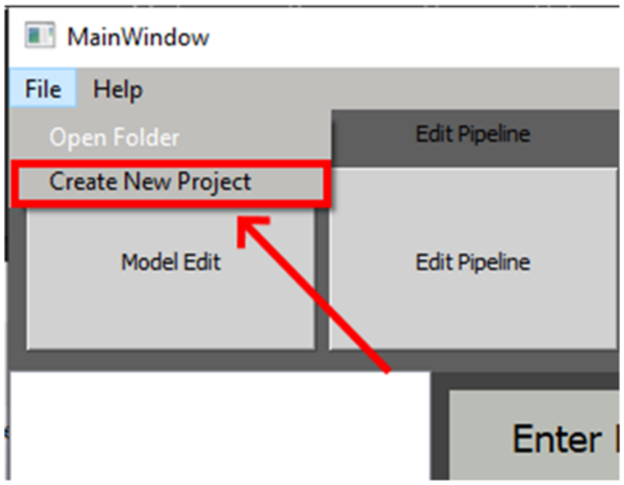
1. Once all object and object types are labled, export the annotations as a .CSV file.


United Nations Global Platform
Home
Project
Annotation
View
Help
Region Shape
Project
Name: via_project_14Feb2024_14h
All files
regular expression
[1] Image__2024-02-14__13-54-40.jpg
[2] Image__2024-02-14__13-59-37.jpg
[3] Image__2024-02-14__13-59-49.jpg
[4] Image__2024-02-14__14-00-00.jpg
[5] Image__2024-02-14__14-01-01.jpg
[6] Image__2024-02-14__14-01-13.jpg
[7] Image__2024-02-14__14-01-21.jpg
[8] Image__2024-02-14__14-01-29.jpg
Export Annotations (as csv)
Export Annotations (as json)
Export Annotations (COCO format)
Import Annotations (from csv)
Import Annotations (from json)
Import Annotations (COCO format)
Preview Annotations
Download as Image
Add Files
Add URL
Remove

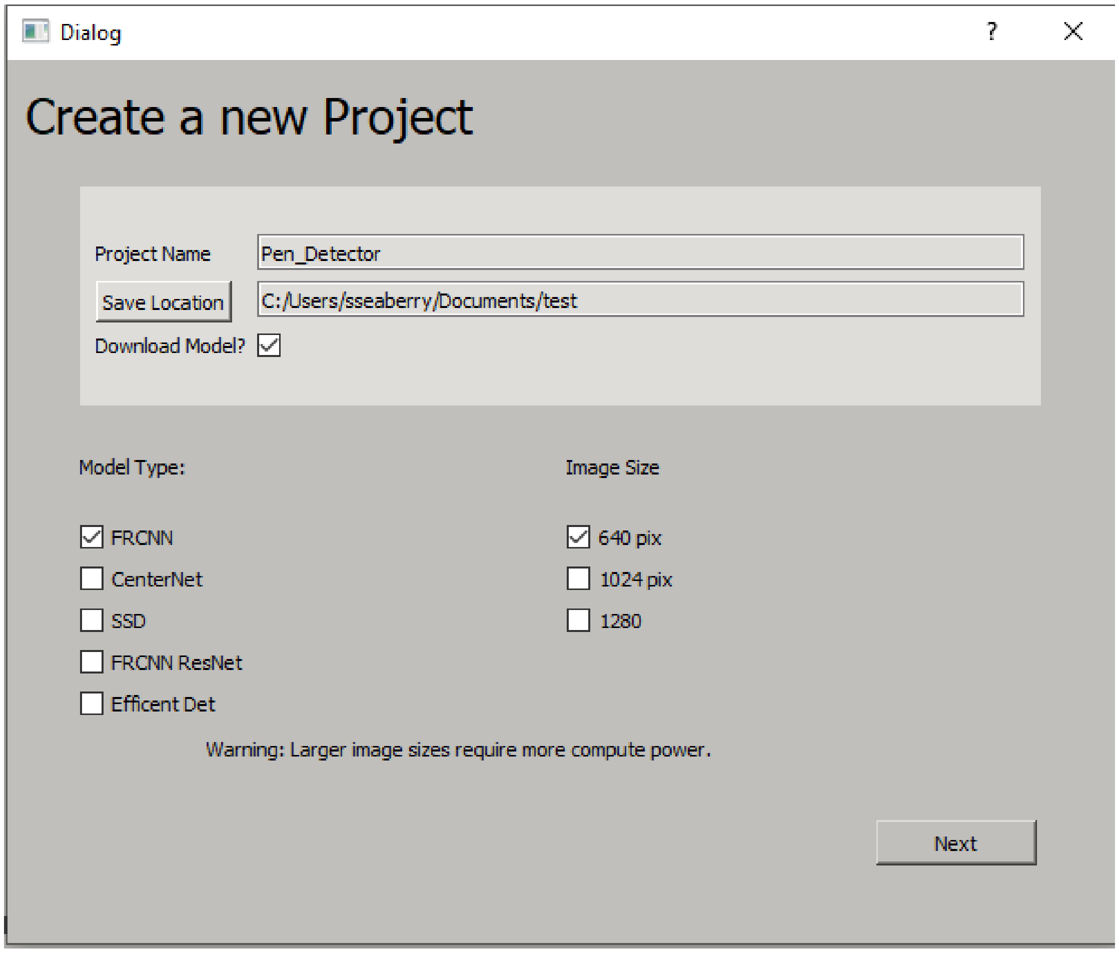
Creating a New Project

2. Launch application
3. Go to File->Create New Project





4. Once the create project dialog opens enter a project name and select a save location for your project.
    a. If desired detection models can be downloaded directly from the create project page. This is the suggested method of downloading tensorflow models.



Create a new Project

Project Name: Pen_Detector

Save Location: C:/Users/sseaberry/Documents/test

Download Model? ✓

Model Type:

- ✓ FRCNN
- ☐ CenterNet
- ☐ SSD
- ☐ FRCNN ResNet
- ☐ Efficent Det

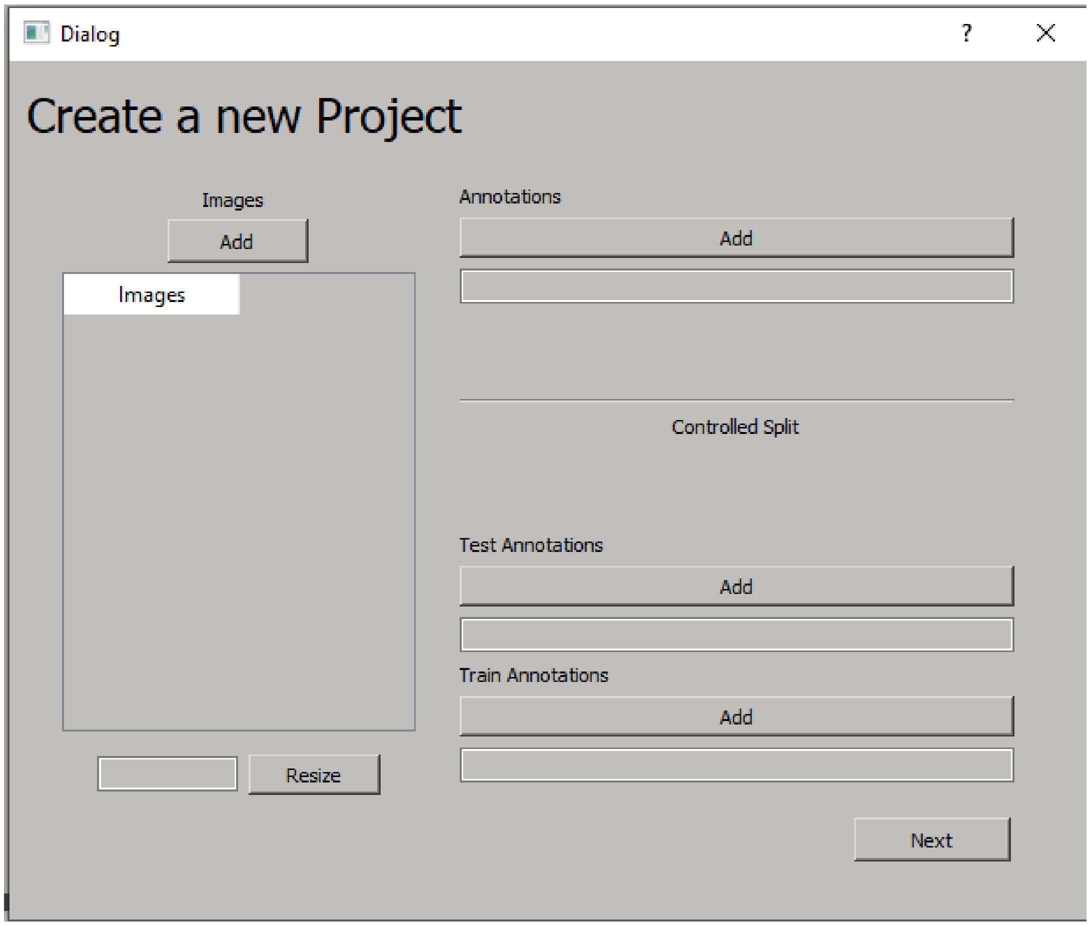
Image Size

- ✓ 640 pix
- ☐ 1024 pix
- ☐ 1280

Warning: Larger image sizes require more compute power.

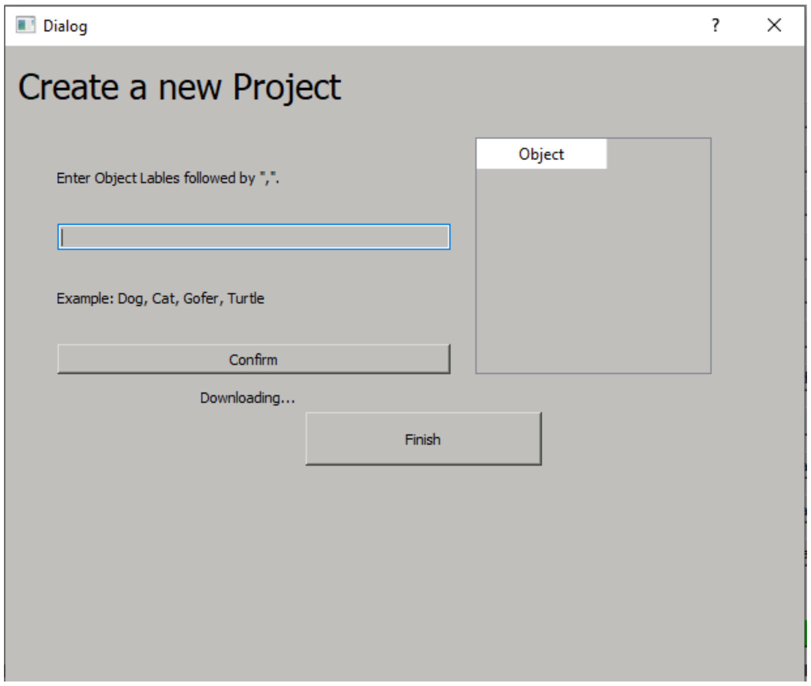
5. Click "Next"
6. The next page loads all user-created files into the project folder. Enter images and annotations (From VGG). If images are not the same size as the selected model image size use the resize option to resize all images. Note images are expected to be square (ie:

640x640 or 1024x1024) for training to be successful.

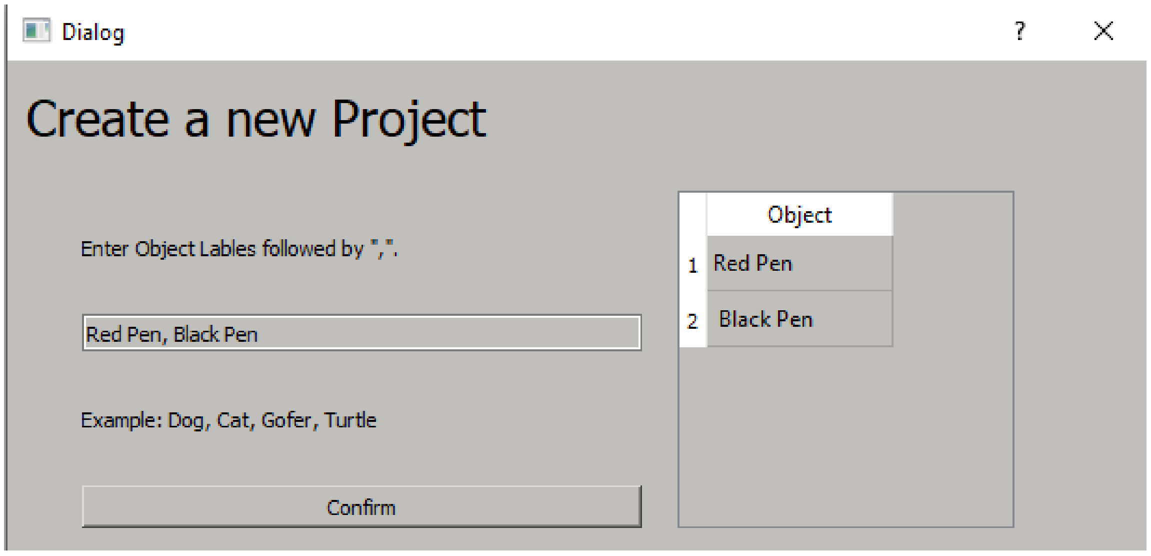


   a. Annotations/images will be split into two groups: Test and Train. This split is done randomly and is not equal. The training dataset will contain 80% of the entire data and the testing will contain the other 20%.
   b. The controlled split option allows users to define a different split percentage and keep train/test datasets separate. This is done by creating two separate annotation files, one containing annotations to train on and the other to test on.

7. Once images and annotations have been provided press “Next” to progress to the next page.


Dialog
? ×
Create a new Project
Enter Object Lables followed by ",".
Example: Dog, Cat, Gofer, Turtle
Confirm
Downloading...
Finish
Object

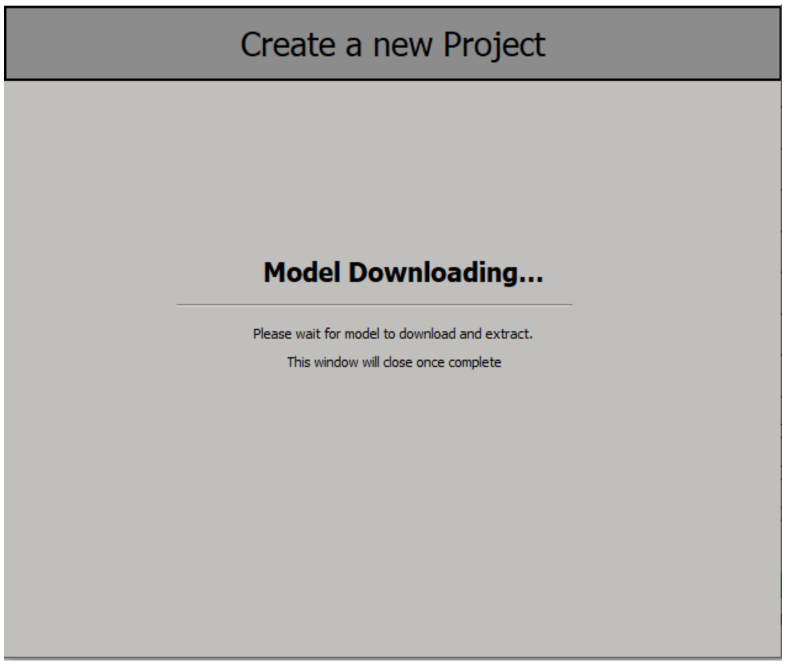
8. Type all lables (these are the attribute names from the annotation step) followed by a comma.
    a. Make sure to type the attribute names exactly the same as during the annotation process. This includes capital letters!
9. Once all labels have been added press "Confirm" and they will appear in the right side table. If satisfied press "Finish".


Dialog
? ×
Create a new Project
Enter Object Lables followed by ",".
Example: Dog, Cat, Gofer, Turtle
Confirm

Red Pen, Black Pen

	Object
1	Red Pen
2	Black Pen

10. If a model has been selected to download this page will appear after pressing finished. The model takes time to download and extract, once this is completed the window will close.



Create a new Project

Model Downloading...

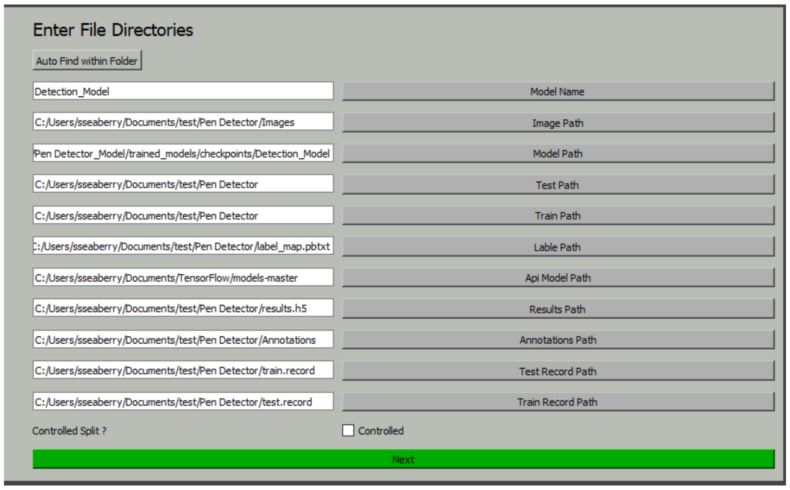
Please wait for model to download and extract.

This window will close once complete

11. The project will automatically be loaded once the window closes.

Training a Model

1. With all the model paths filled in. Press "Next" to advance to the next page.
    a. Note: If using controlled annotations splitting ensure the "Controlled Split" checkbox is checked.



Enter File Directories

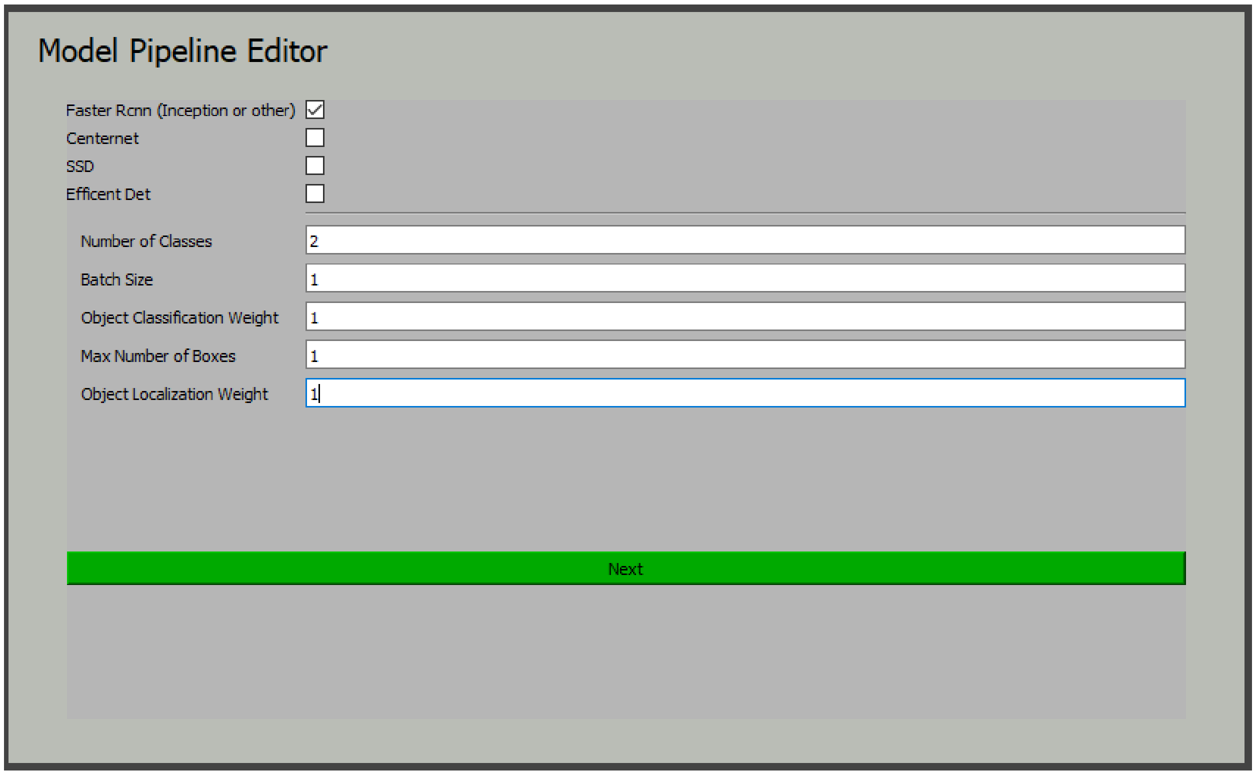
Auto Find within Folder

Detection_Model	Model Name
C:/Users/sseaberry/Documents/test/Pen Detector/Images	Image Path
Pen Detector_Model/trained_models/checkpoints/Detection_Model	Model Path
C:/Users/sseaberry/Documents/test/Pen Detector	Test Path
C:/Users/sseaberry/Documents/test/Pen Detector	Train Path
:/Users/sseaberry/Documents/test/Pen Detector/label_map.pbtxt	Lable Path
C:/Users/sseaberry/Documents/TensorFlow/models-master	Api Model Path
C:/Users/sseaberry/Documents/test/Pen Detector/results.h5	Results Path
C:/Users/sseaberry/Documents/test/Pen Detector/Annotations	Annotations Path
C:/Users/sseaberry/Documents/test/Pen Detector/train.record	Test Record Path
C:/Users/sseaberry/Documents/test/Pen Detector/test.record	Train Record Path
Controlled Split ?	☐ Controlled

Next

2. The next page gives users the ability to edit the models parameters to fit their specific model and use case.
    a. Select the correct model type (here FRCNN).
    b. Number of classes: number of independent annotations.

c. Batch size: number of images to train on before updating weights/biases. If training on a single GPU/CPU set this value to 1-6. (NOTE: larger numbers will need more computation resources. If getting a OOM error when training lower this value.)
d. Object Classification Weight/Object Localization Weight: Set these to values floating point numbers above 1. If classification accuracy is more important set this value larger than the localization and vise versa.
e. Number of boxes: Set to the largest number of boxes in a single image. NOTE: If training on images with > 100 objects in a single image, see fine-tuning section.


Model Pipeline Editor

Faster Rcnn (Inception or other) ✓
Centernet ☐
SSD ☐
Efficent Det ☐

Number of Classes	2
Batch Size	1
Object Classification Weight	1
Max Number of Boxes	1
Object Localization Weight	1

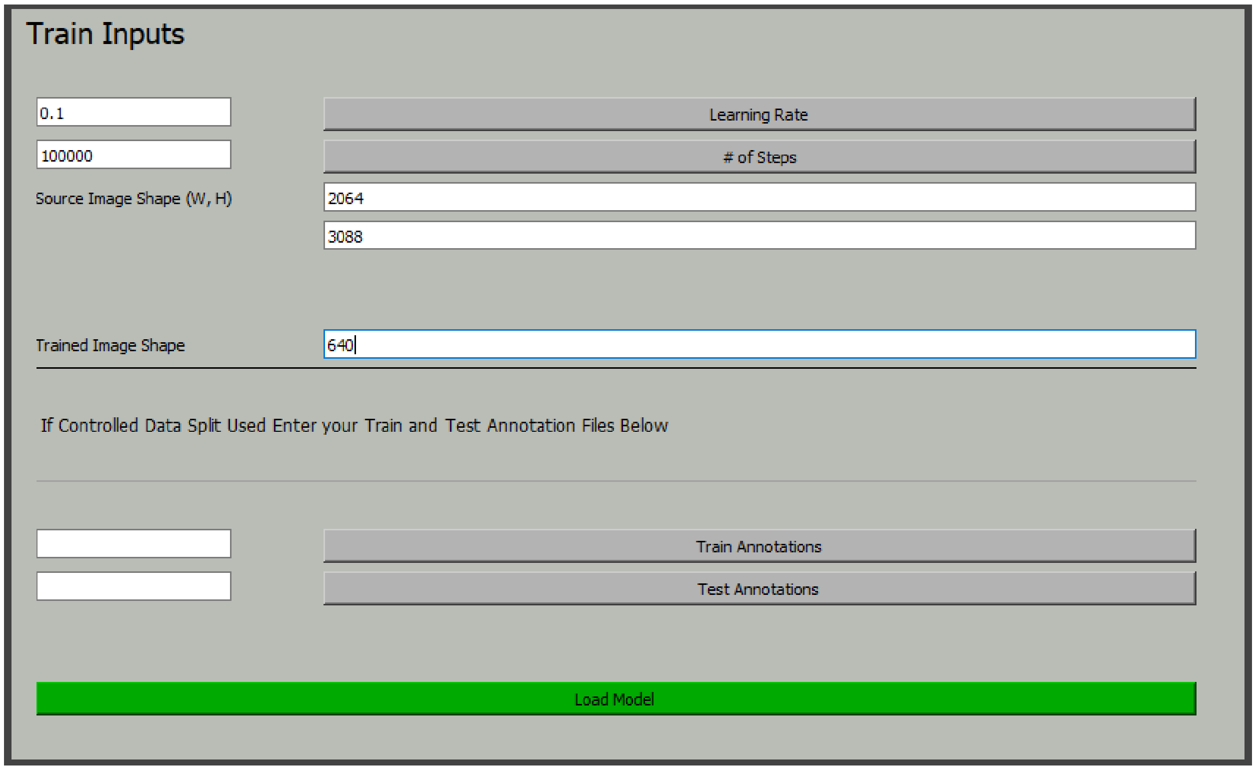
3. Press "Next" to progress to the next page.
4. On the "Train Inputs" page:
    a. Enter a learning rate between 1 and 0.001. This value has a large impact on the model's training losses. Too large and losses will increase, too small and they will decrease too slowly/overfit. It is suggested to start at 0.1 and decrease by half if the model loss is increasing.
    b. # of steps: Set to value above 100,000 idea is 300,000+.
    c. Source image shape: Used to correctly scale annotations, set these values to the Width/height of the original images used to create annotations.
        i. These values will auto-import like the model paths
        ii. To find these values without auto-import:
            1. Right-click on any image originally used during the annotation process.
            2. Select "Properties" from the menu.
            3. Go to the "Details" tab.

4. Scroll down to the "Image" section where the height and width dimensions will be listed.

d. Trained image shape: Size of model training images. Selected during the project creation process.

e. If using a controlled split:

   i. Double-click and drag training/test annotation files from the left-hand directory viewer into their respective boxes.



Train Inputs

0.1	Learning Rate
100000	# of Steps
Source Image Shape (W, H)	2064
	3088
Trained Image Shape	640

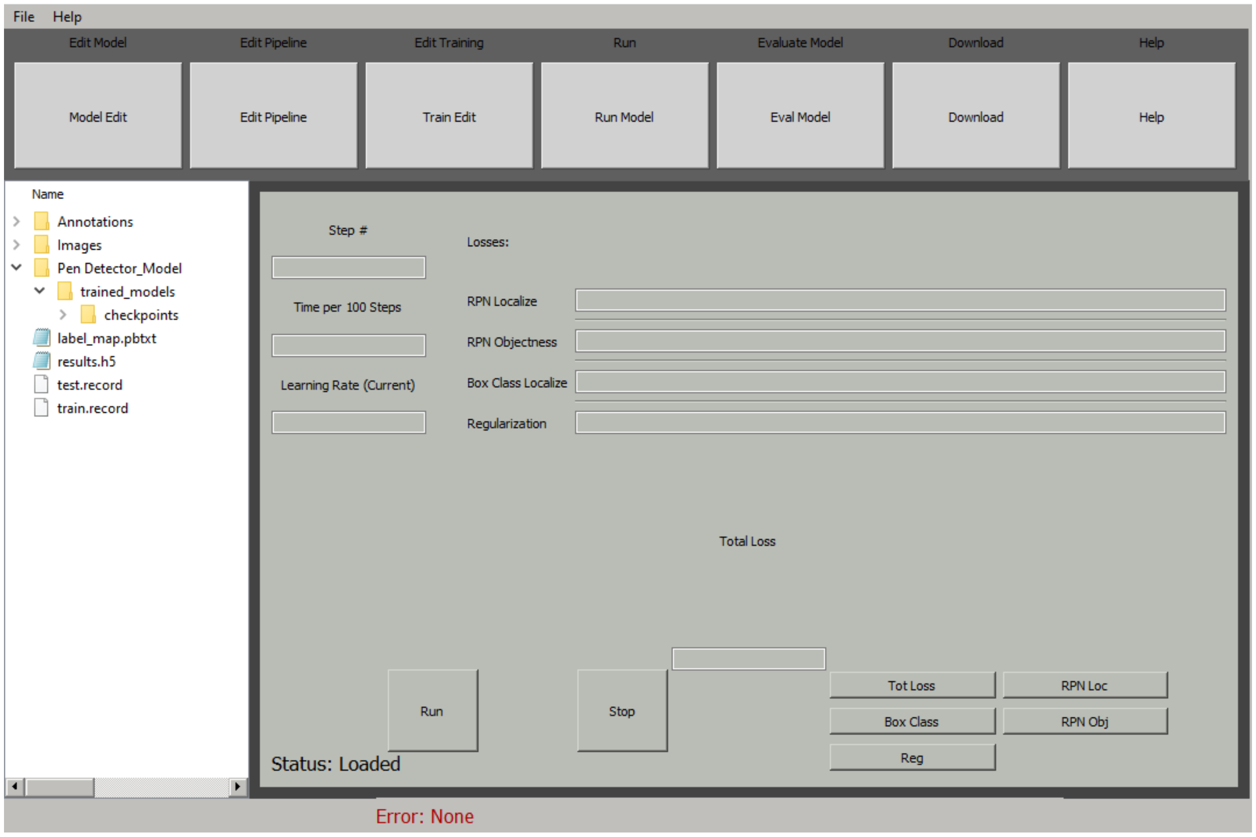
If Controlled Data Split Used Enter your Train and Test Annotation Files Below

	Train Annotations
	Test Annotations

12. Press "Load Model".

    a. If model loading is successful, the Error display will show "None".

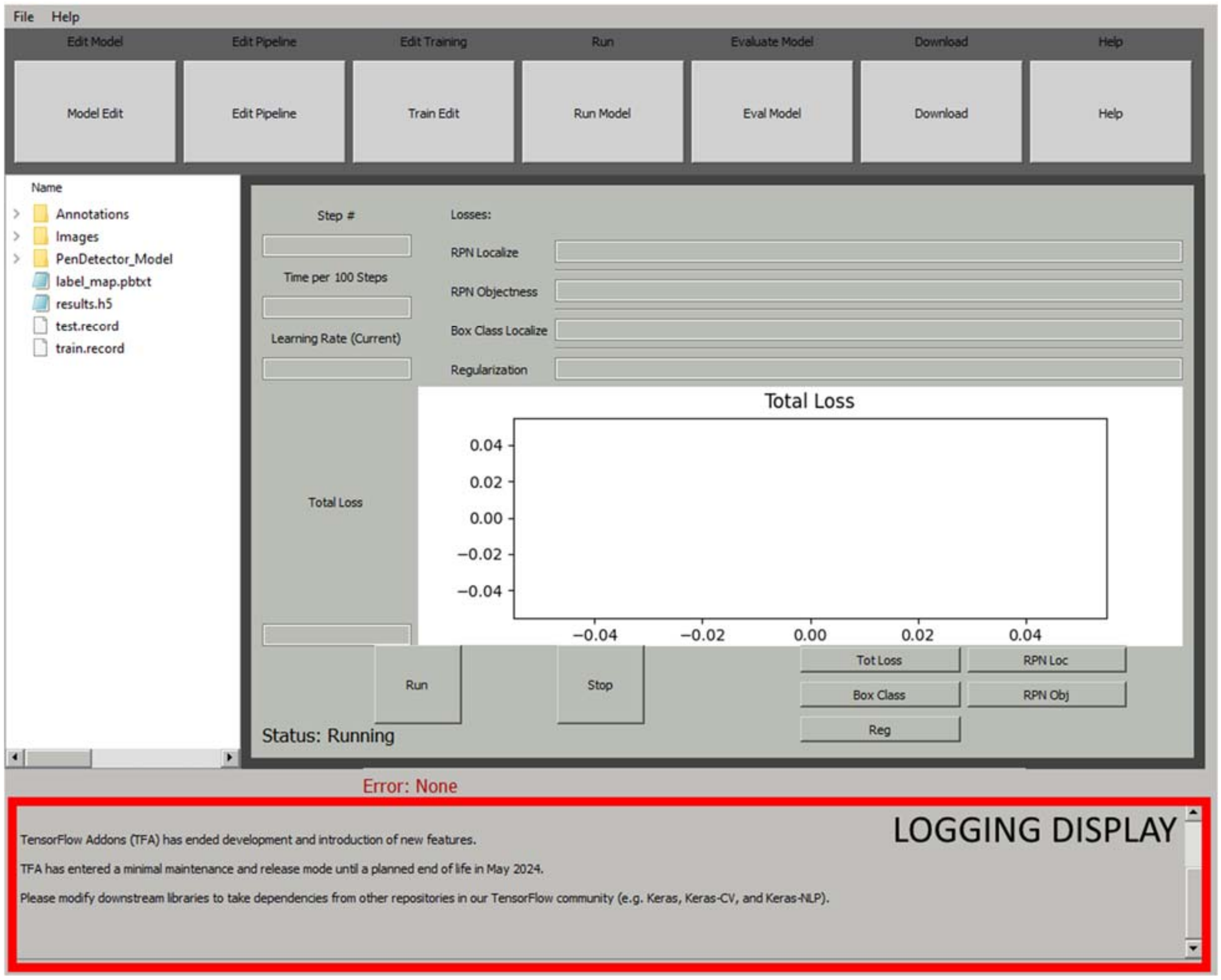
13. From the top taskbar select "Run Model"



Status: Loaded

Error: None

14. If there are no errors and the Status reads “Loaded”, press “Run”
15. If successful the Status will read “Running” and logging output will start to roll out on the log display below.

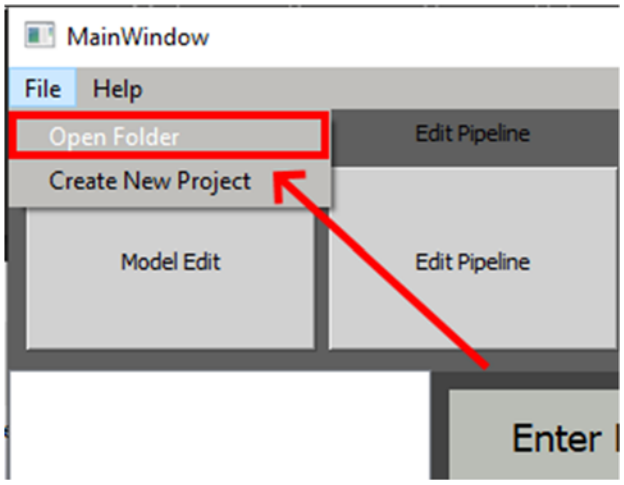


```
TensorFlow Addons (TFA) has ended development and introduction of new features.
TFA has entered a minimal maintenance and release mode until a planned end of life in May 2024.
Please modify downstream libraries to take dependencies from other repositories in our TensorFlow community (e.g. Keras, Keras-CV, and Keras-NLP).
```

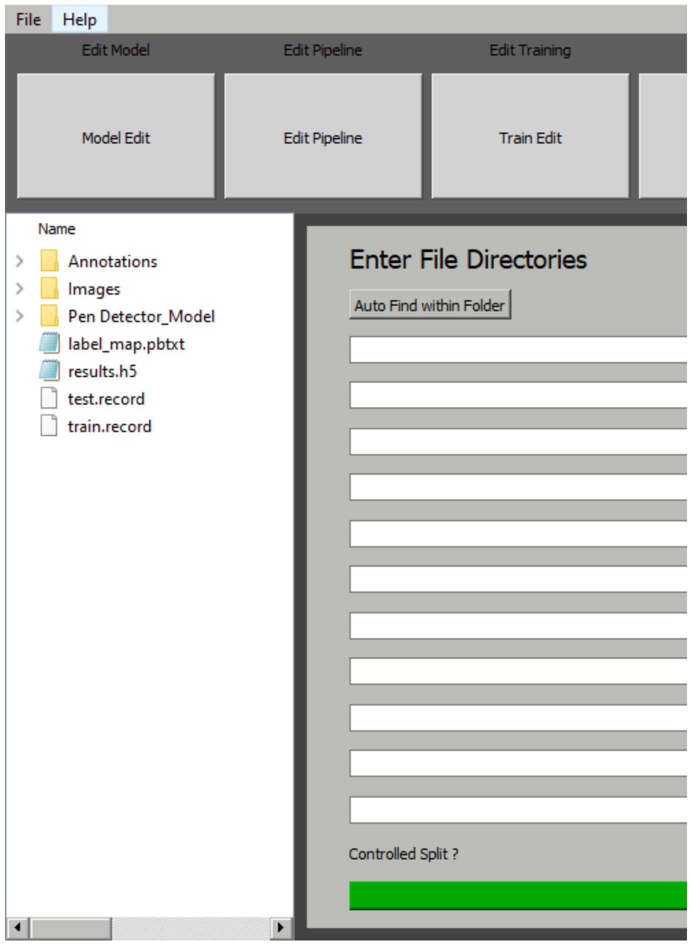
During the models training the losses are updated and saved so that graphs of each loss can be viewed in real time. NOTE: If training on a single GPU/CPU loss values can take a long time to update due to slow model training speeds. It is typical to see +1h of delay between loss updates. This is VERY model/image size dependent.

Importing a project

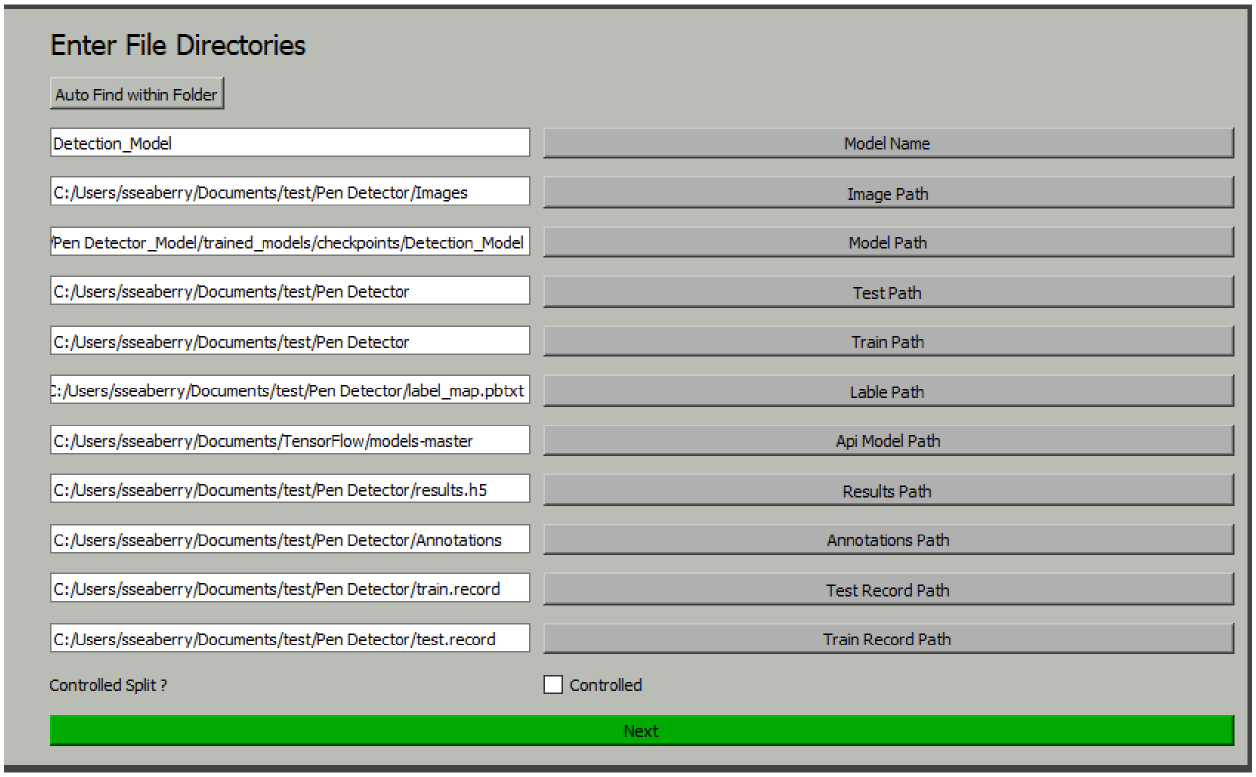
1. Click on File->Open Folder



2. Open the project folder.
    a. The project directory will then be displayed in the left hand table.



3. With a project loaded press "Auto Find within Folder". This will automatically load all model training parameters.


Enter File Directories

Auto Find within Folder

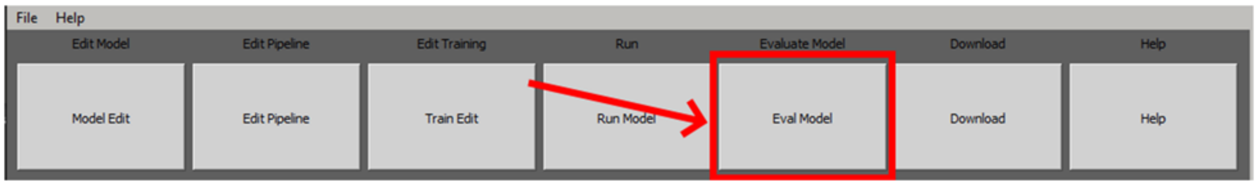
Detection_Model	Model Name
C:/Users/sseaberry/Documents/test/Pen Detector/Images	Image Path
/Pen Detector_Model/trained_models/checkpoints/Detection_Model	Model Path
C:/Users/sseaberry/Documents/test/Pen Detector	Test Path
C:/Users/sseaberry/Documents/test/Pen Detector	Train Path
:/Users/sseaberry/Documents/test/Pen Detector/label_map.pbtxt	Lable Path
C:/Users/sseaberry/Documents/TensorFlow/models-master	Api Model Path
C:/Users/sseaberry/Documents/test/Pen Detector/results.h5	Results Path
C:/Users/sseaberry/Documents/test/Pen Detector/Annotations	Annotations Path
C:/Users/sseaberry/Documents/test/Pen Detector/train.record	Test Record Path
C:/Users/sseaberry/Documents/test/Pen Detector/test.record	Train Record Path

Controlled Split ? ☐ Controlled

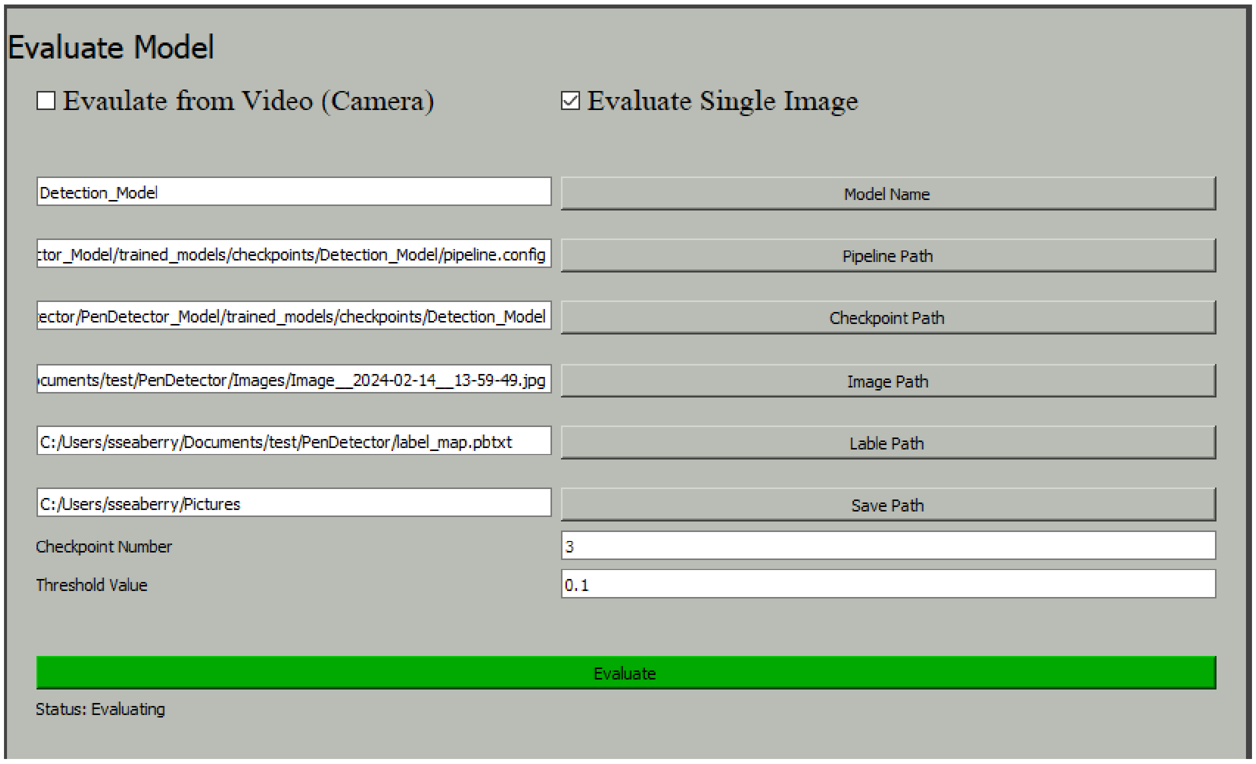
Next

Evaluating a Model

Click on “Eval Model” in the top taskbar



The model eval page:


Evaluate Model

☐ Evaulate from Video (Camera) ☑ Evaluate Single Image

Detection_Model	Model Name
:tor_Model/trained_models/checkpoints/Detection_Model/pipeline.config	Pipeline Path
ector/PenDetector_Model/trained_models/checkpoints/Detection_Model	Checkpoint Path
cuments/test/PenDetector/Images/Image__2024-02-14__13-59-49.jpg	Image Path
C:/Users/sseaberry/Documents/test/PenDetector/label_map.pbtxt	Lable Path
C:/Users/sseaberry/Pictures	Save Path
Checkpoint Number	3
Threshold Value	0.1

Status: Evaluating

First choose to either evaluate the model of a single static image (Evaluate Single Image) or to evaluate the model on a constantly updated video (Evaluate from Video).

As before on the Run page type the model's name. Note if the project was set up using the Create Project function this will be “Detection_Model”.

Pipeline path should point to the pipeline.config file in the model's folder.

Checkpoint path should point to the folder containing the checkpoints.

Image path: If evaluating from a single image enter the path to the image. If evaluating from camera this entry can be set to anything.

Label path should point to the labelmap.pbtxt of the model.

Save path location to save images too.

Checkpoint number is an integer value corresponding to the checkpoint of the model.

Threshold value is a floating-point number between 1 and 0 corresponding to the accepted certainty levels for detected objects.

Evaluation from Video will pick up video from whatever camera is attached to the user's machine. If no camera is detected the evaluation will terminate unexpectedly.

Fine Tuning

Fine-tuning is done by editing the models pipeline.config file. For a full explanation of the pipeline file and all its options visit:
https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/configuring_jobs.md

If there are more than 100 objects per image the pipeline.config must be edited to include this line in the EVAL INPUT READER section:

max_number_of_boxes: 500 (or whatever is the max number of objects in a single image)

If training on images that are not similar to those in the COCO dataset users may want to start training a model from scratch. To do so, remove the FINE_TUNE_CHECKPOINT lines in the pipeline.config file.

If the objects/images are not subjected to large variations in lighting, or coloring users may want to train without data augmentation. To do so; remove all the data augmentation sections from the pipeline.config.

Model Descriptions

FRCNN

FRCNN is one of the best models for accuracy, beating SSD, Centernet and EffDet when evaluated over the coco dataset. This accuracy does come at a cost; FRCNN is a SLOW model meaning evaluation time for a single image will be much longer when compared to other models. Chose this model type when objects are very hard to define or if accuracy is paramount and execution time is not.

SSD

SSD is a very fast and quite small model, commonly used for embedded applications and detecting from a video. This speed does have its pitfalls when it comes to the model's overall accuracy. Choose this model if speed and memory size are more important than accuracy and/or if the objects are very easy to define.

Centernet

Centernet is about as fast as SSD but is a larger model. If SSD is not accurate enough choose this model. The execution times are slightly slower but the models ability to locate hard-to-define objects is better.

EffDet

EffDet is slower than Centernet and SSD but faster than FRCNN, with accuracies higher than Centernet and just below FRCNN. Choose EffDet if Centernet is not accurate enough or if execution times need to be slightly faster than that of FRCNN.

Setup Without Using Create Project

Model Training:

To train the model, several parameters must be given before the program can run. Most of these are paths to files or folders.

- Model Name, STRING corresponding to saved model folder name
- Image Path, FOLDER PATH containing all images to be used for both training and testing
- Model Path, FOLDER PATH to downloaded model folder
- Config Path, FILE PATH pointing to pipeline.config file
- Test Path, FOLDER PATH to location of test.record
- Train Path, FOLDER PATH to location of test.record
- Lable Path, FILE PATH pointing to lablemap in .pbtxt form
- Apimodel Path, FOLDER PATH to tensorflow object detection models folder
- Checkpoint Path, FOLDER PATH to selected models checkpoint folder (rename this from default to checkpoint0 if not already called checkpoint0)
- Result Path, FILE PATH pointing to models .h5 file (Can be located anywhere and can be created by user)
- Annotations Path, FILE PATH pointing to annotations .txt file (use vgg annotator (https://annotate.officialstatistics.org/) and convert .csv to .txt by changing thenendswitch using F2 -> rename DO NOT open in excel and save to .txt)
- Test Record, FILE PATH pointing to models test.record file (Can be located anywhere and can be created by user)
- Train Record, FILE PATH pointing to models train.record file (Can be located anywhere and can be created by user)
- controlled, set to 1 if user needs controlled spilt of train and test data select 0 if random split is OK

Train page

Learning rate:

The learning rate corresponds to how much each tensor is changed each epoch. Its is recommended to use a value between 1 and 0.1. Setting the learning rate to a larger value will result in a faster convergence but may cause the model to converge on a non-optimal solution. A smaller value will give a slower convergence but will result in a more optimal solution. Of course, this depends greatly on the model and objects you are trying to locate, but this is true for the general case.

It is also recommended to use a variable learning rate. This can be achieved by using a stepped learning rate or cosine decay learning rate. Both options decrease the learning rate as the model progresses. Note that if cosine or another uncontrolled variable learning rate can cause model to stagnate if the initial learning rate is set to low as it will tend to zero and will not recover unless the model is reset

# of steps

Minimum for the model to properly converge is 150,000 – 200,000 ideal is 300,000+. With less training images the number of steps needed to converge will increase.

Test & Train Annotations

Input the split train and test annotations. These must be split by image ie: images 1-7 are used for training and 8-10 are for evaluation, test annotations will only contain annotations for images 1-7 and 8-10 only for training annotations. Failure to split annotations properly will result in a sub-optimal convergence.

Creating Annotations

To create annotations for object detection use https://annotate.officialstatistics.org/.
Steps:

1. Create annotations for all objects in each image
    a. If there is more than one object that needs to be classified, make sure to use multiple attributes and use the proper attribute label for the object.
        i. Ie: Trays and setters if you create a bounding box around a setter make sure in VGG, that the box is linked to the setter attribute label.
2. Export the complete annotations as a csv file.
    a. Top bar -> Annotations -> Export Annotations (as csv)

Evaluation

Paths are the same as they are in model edit. Save location needs to point to an EXISTING image file of the same size (640x640 for resnet 640).

Checkpoint number: INT value

The checkpoint that the model should be evaluated at. Commonly this is the most recent checkpoint. See file structure to find where to obtain the checkpoint number from.

Threshold value: FLOAT (MUST BE BETWEEN 0-1)

This input thresholds the minimum confidence for an object to be classified. Suggested values are between 0.5 and 0.7. As the threshold value tends to 1 there will be less and less objects classified.

File structure

```
Modelname_Folder
        |-------> Pipeline.config FILE
        |-------> ckpt-*(Checkpoint files)
        |-------> checkpoint0 FOLDER
                |-------> checkpoint
                |-------> ckpt-0
        |-------> eval FOLDER
                |-------> events.out.* FILE(s) Filled if eval has been run on model
        |-------> train folder
                |-------> events.out.* FILE(s) Filled at regular intervals
```

```
|-------> Saved_model FOLDER
          |-------> Saved_model.pb FILE
|-------> Variables FOLDER
          |-------> variables.* FILE(s)
```

Downloading of new models

Downloadable models can be found here:
https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

Steps:

1. Right Click on model name
2. Click “copy link address”
3. Paste into form

Image shapes

Most of the time users will want to annotate using full-resolution images to be able to accurately define the bounding boxes, and then down-sample the images for object detection training. If original scale images are used there can be an overallocation of resources, resulting in crashes. Due to this, there are fields for Source images (full-resolution) and Train Images (down-sampled) in the GUI. Note that incorrect inputs here will have catastrophic consequences on the detection algorithm, either with crashing or bounding boxes being in incorrect locations.

Example inputs:

- Source images, input field 1 = width
  - input field 2 = height
- Train image, input field = width and height (assumed square image)
  - **** SINGLE VALUE HERE (ie 512x512 image enter just 512)