Ryan Filgas
Program 4
Computer Vision

Program four covered the Grad CAM DEL-Based Saliency Algorithm

For this program I used a pre-trained VGG16 CNN to generate class discriminative saliency maps. The algorithm uses the gradient information from the final convolutional layer to generate the maps.

Step 1: Import images and preprocess them using pre_process_input from the tensoflow vgg16 library. This was easier said than done as I initially tried to batch the images and was struggling to find the right syntax for preprocessing. I saved the model locally to be loaded in later.

Step 2: Once the import step was complete, I passed the image set through the vgg16 model and got the predictions for the image set to include the top three for each class.

These are my top 3 initial predictions for each of the 5 images. These predictions changed slightly once I changed my assignment to reflect the initial gradient tape function we used. New predictions will be indicated in the attached plots. They're the same for the most part.

```
[[('n04037443', 'racer', 0.21141012),
  ('n04467665', 'trailer_truck', 0.15876162),
  ('n03930630', 'pickup', 0.12652741)],

 [('n03590841', "jack-o'-lantern", 0.49427733),
  ('n07717410', 'acorn_squash', 0.1939868),
  ('n02797295', 'barrow', 0.09410795)],

 [('n07734744', 'mushroom', 0.9262988),
  ('n12998815', 'agaric', 0.061288718),
  ('n13054560', 'bolete', 0.0048882808)],

 [('n02106030', 'collie', 0.11618046),
  ('n02110806', 'basenji', 0.09162874),
  ('n02129165', 'lion', 0.07576906)],

 [('n02690373', 'airliner', 0.98696554),
  ('n04592741', 'wing', 0.012698307),
  ('n04266014', 'space_shuttle', 0.0001545953)]]
```

These are my initial top 1 indexes for each image. I believe they're the same after with exception to the vehicle which was classified later as a tow truck.

```
[607 404 947 751 231]
```

Ryan Filgas
Program 4
Computer Vision

Step 3: Determine the argmax of the prediction vector from step 2 to get the index of the top 1 predicted class, then extract the last conv layer of the model. Following this I attempted to us K.gradients, and tried several work arounds due to deprecation. I spent several hours trying to learn how to use the new gradient tape function (admittedly I missed information that could have been easy to find). The new gradient tape function takes an input array, a model, and a predicted index. That model is created using model inputs, the conv layer output, and the model output. Once this was discovered things went much more smoothly.

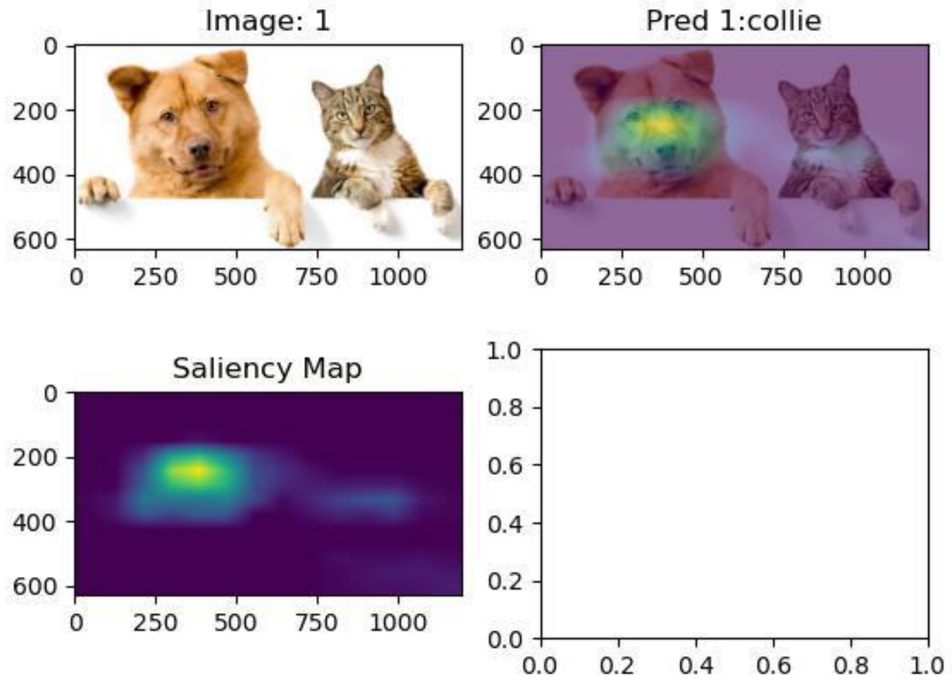
Step 4: Implement equations 1 and 2 from the assignment sheet. These took some doing, but the process was pretty straight forward. The gradient tape function produced our gradients, and to get the weights they were just summed with respect to each feature and divided by 14x14 – the number of filter "pixels". After this for part 2 I multiplied the conv layer output by the alphas from the previous step before summing them and applying a relu to set all negative values to 0.

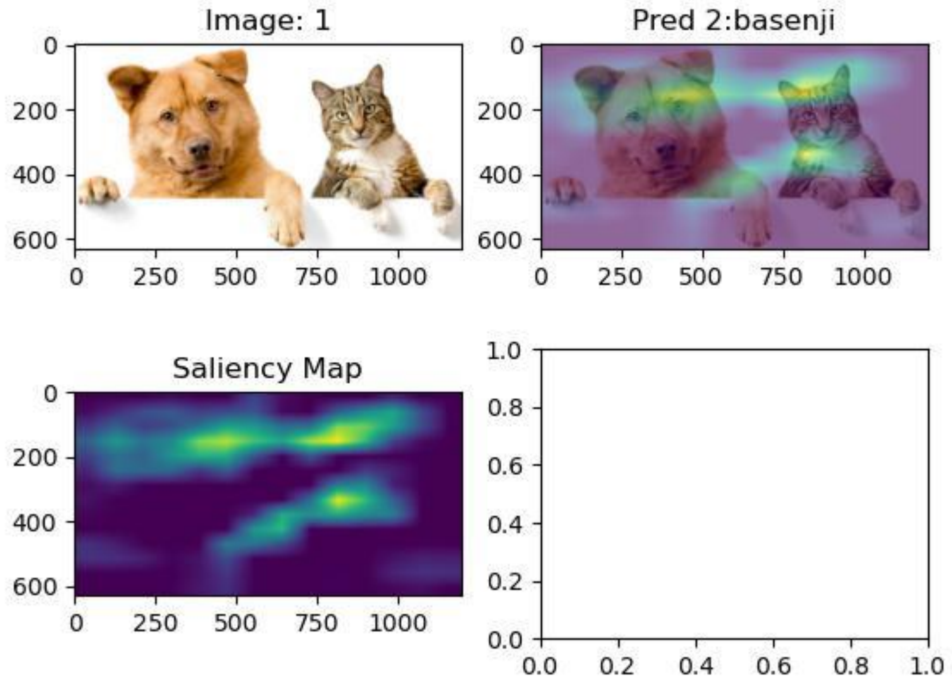
The process of locating the indexes after this was a little bit clunky on my end. I took each value from the top 3 predictions and matched it up to the corresponding index. After this I used a combination of opencv and pyplot to create my visualizations. I tied things up by batching up all of the work for a single image and repeating this process for each image. Results are below.

In general all of the results are expected, however there were a couple eyebrow raisers. The first interesting thing was the airplane, where the saliency map completely misses the main feature of the plane and highlights up in the corner. The second thing that seemed off was the vehicle. It was blocky, so part of this is understandable, but it identified the vehicle as a tow truck. As far as other observations go, the second and third choices for each group tended to be less localized to the actual object in the image.

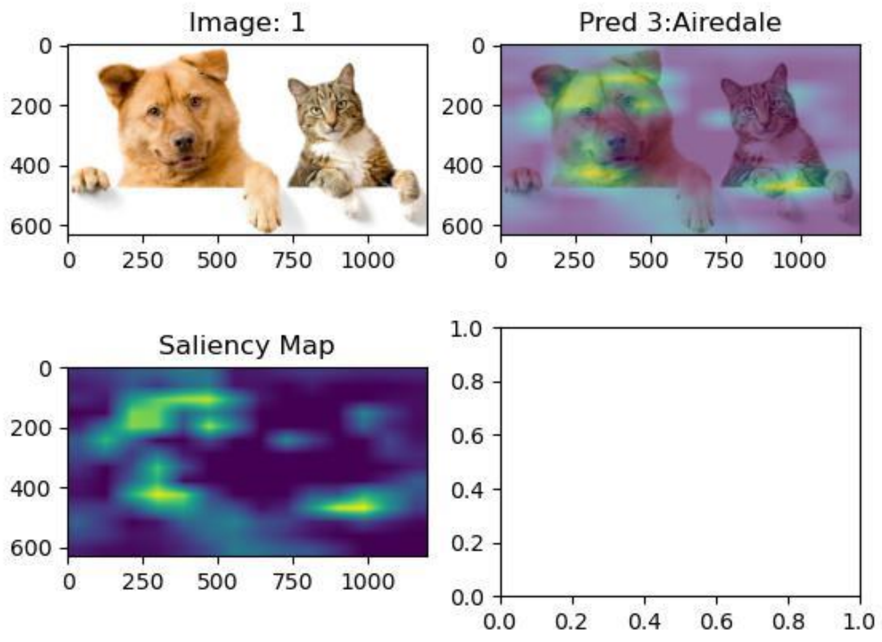
The mushroom classification in particular was interesting as you could see different parts of the mushroom were being picked up on, the same is true of the vehicle. The information gained by doing this mapping can help develop better models later and shed some light on what classifiers are doing.

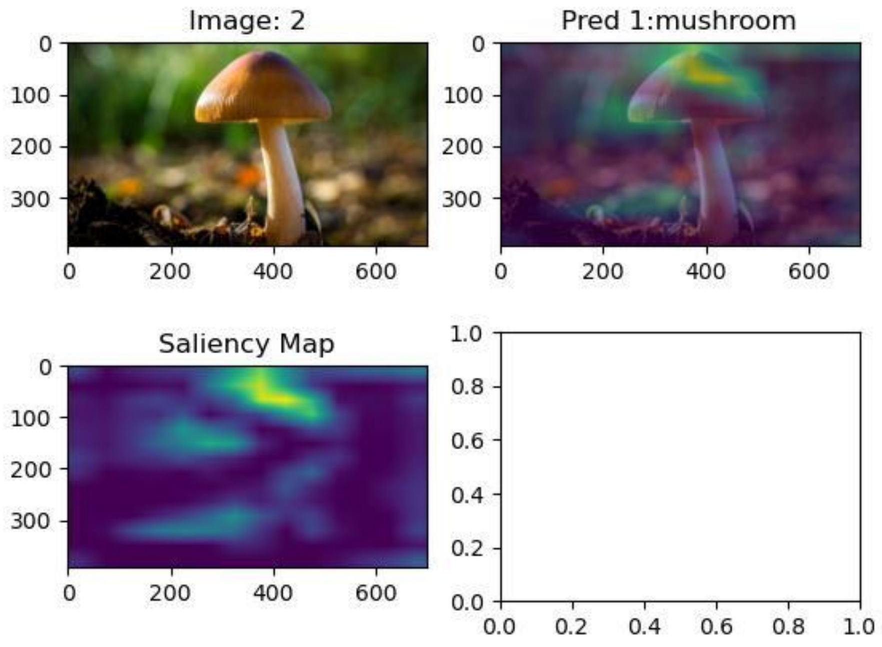
Ryan Filgas
Program 4
Computer Vision



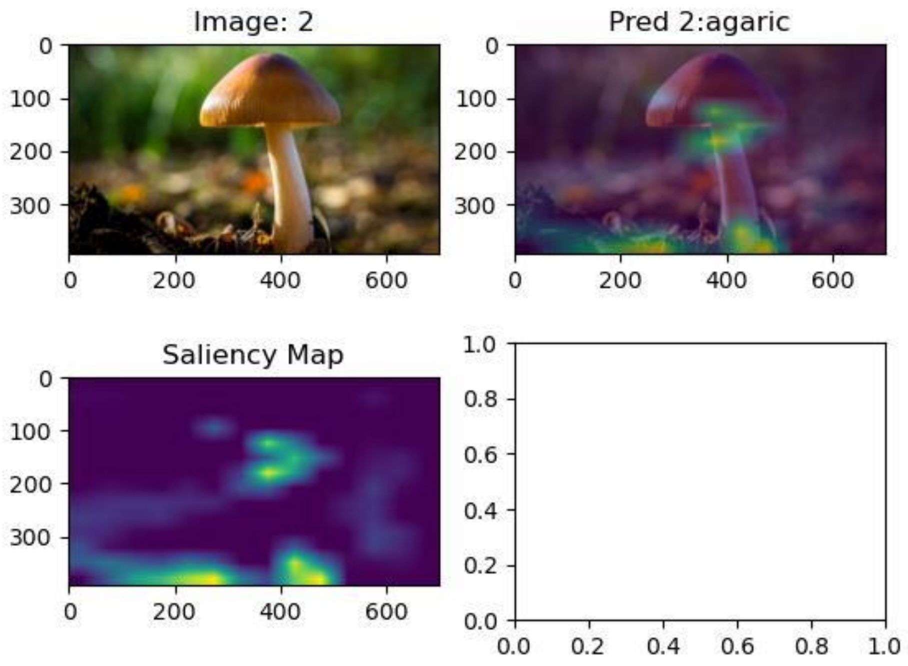


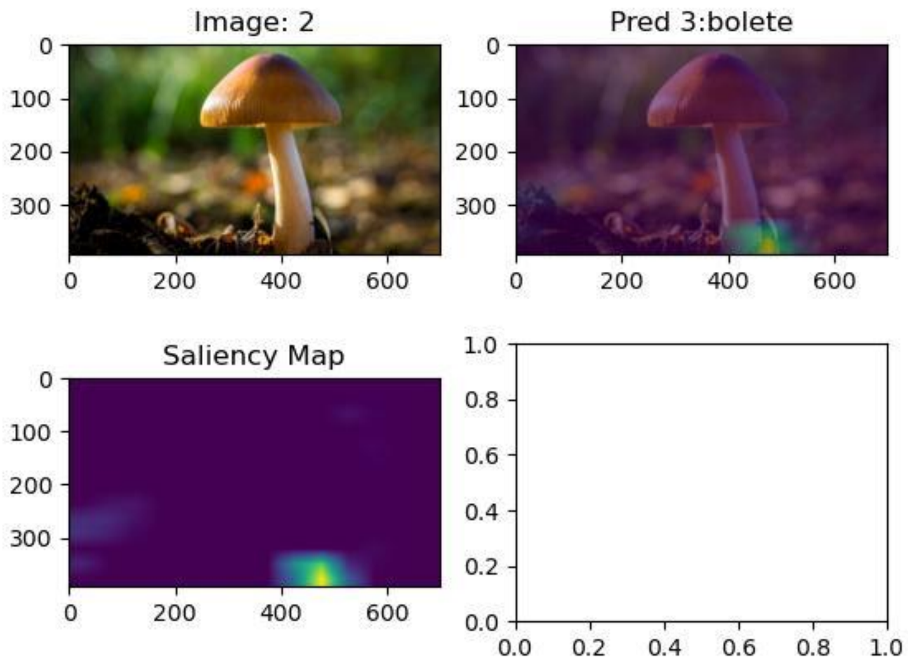
Ryan Filgas
Program 4
Computer Vision



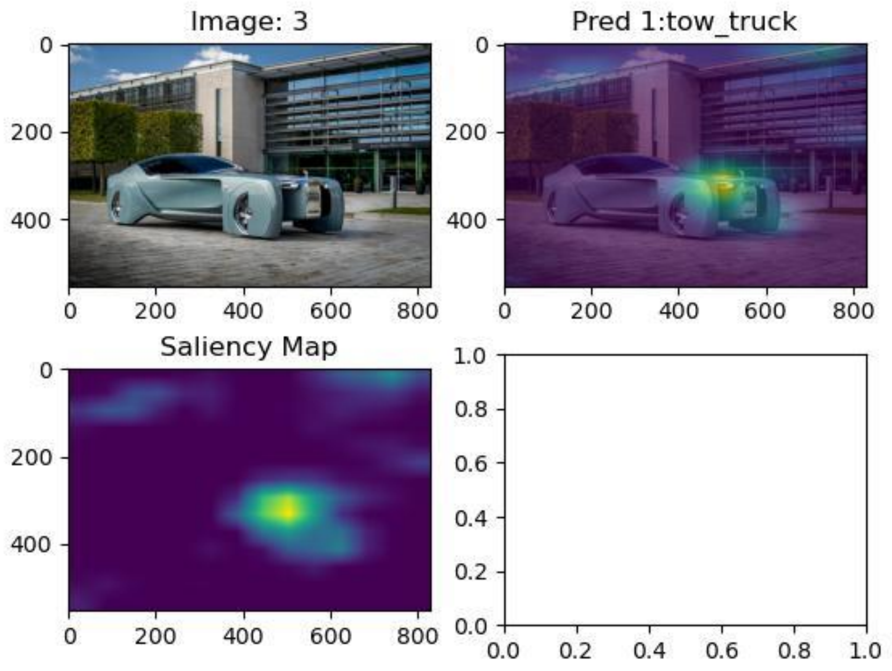


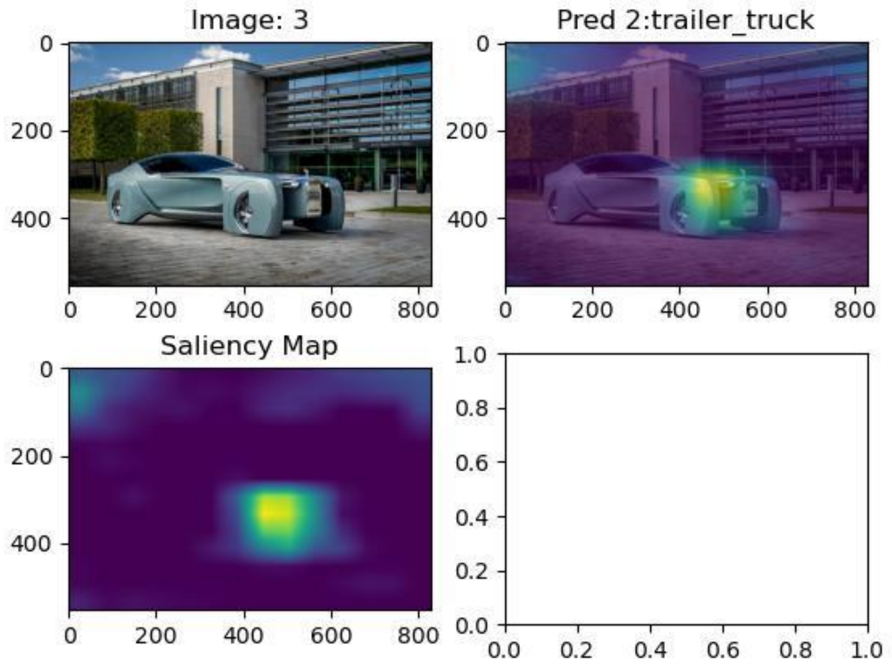
Ryan Filgas
Program 4
Computer Vision



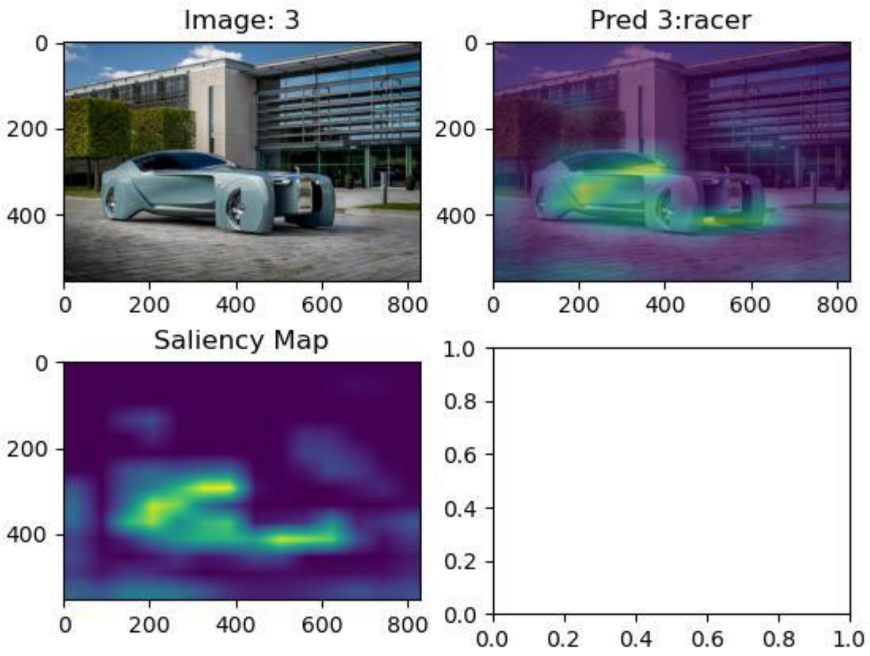


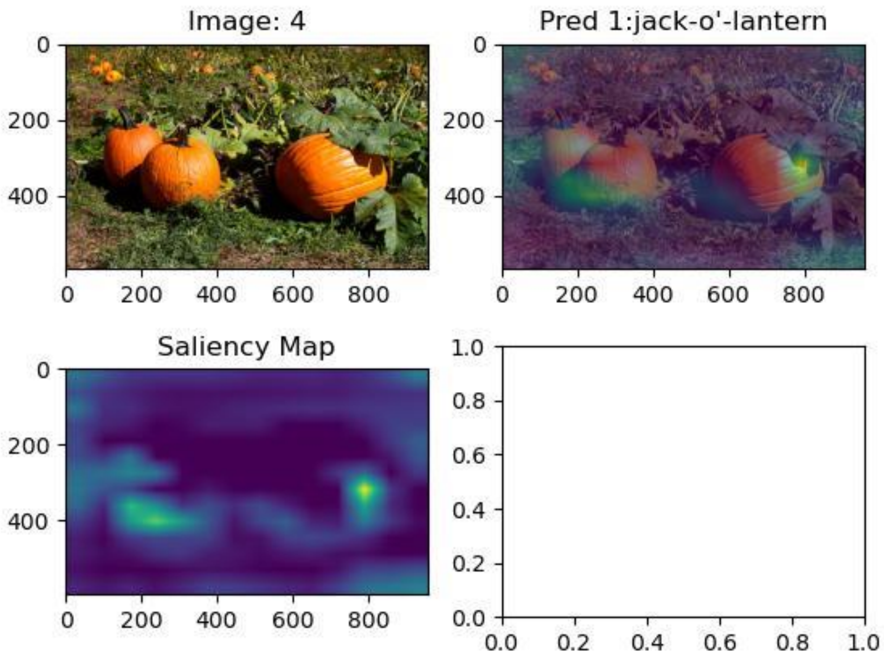
Ryan Filgas
Program 4
Computer Vision



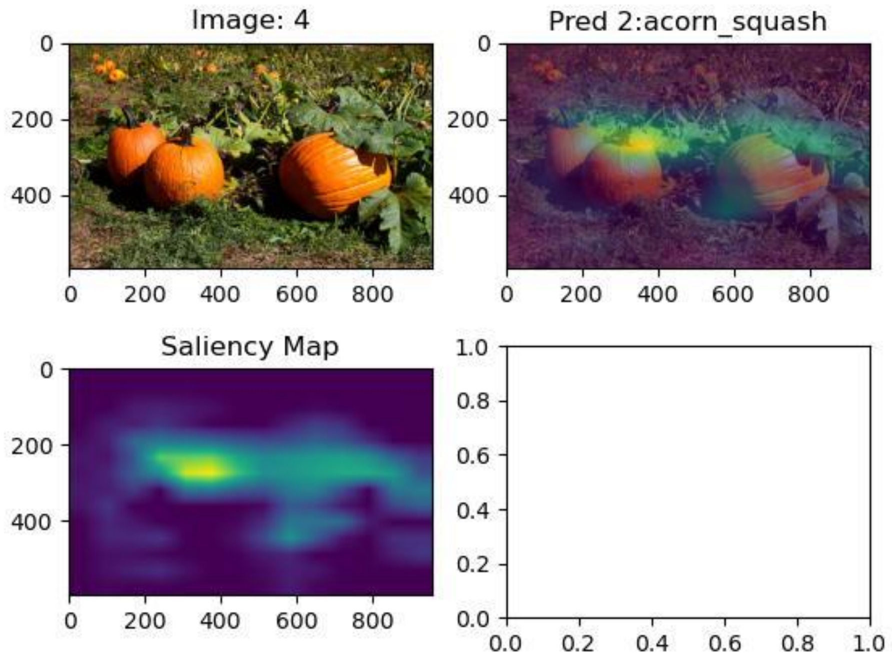


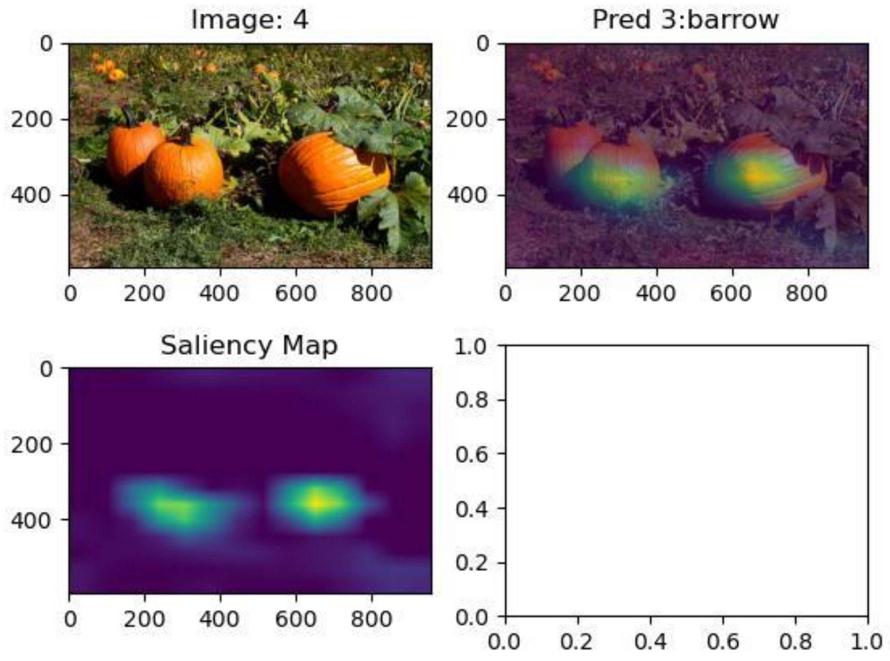
Ryan Filgas
Program 4
Computer Vision



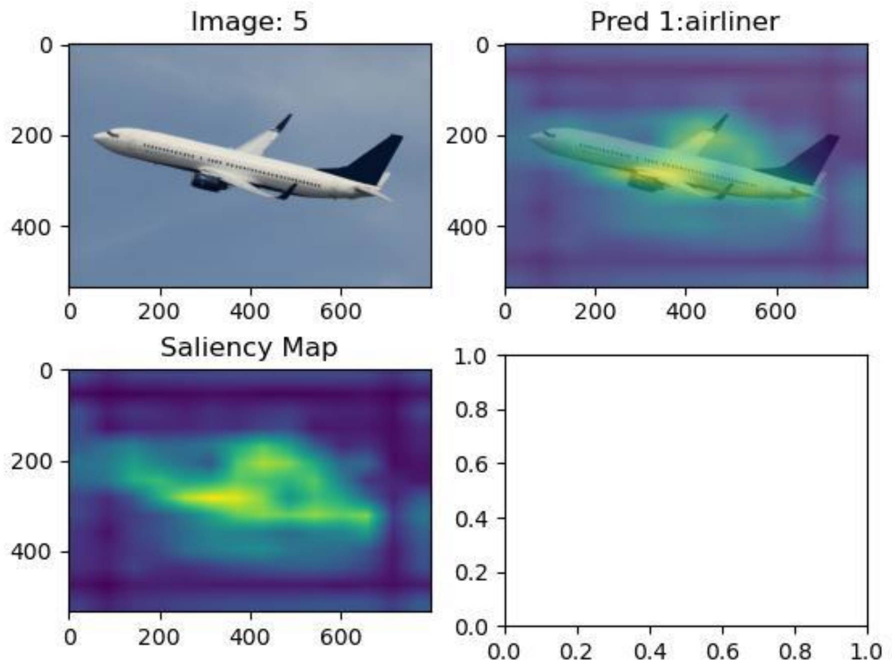


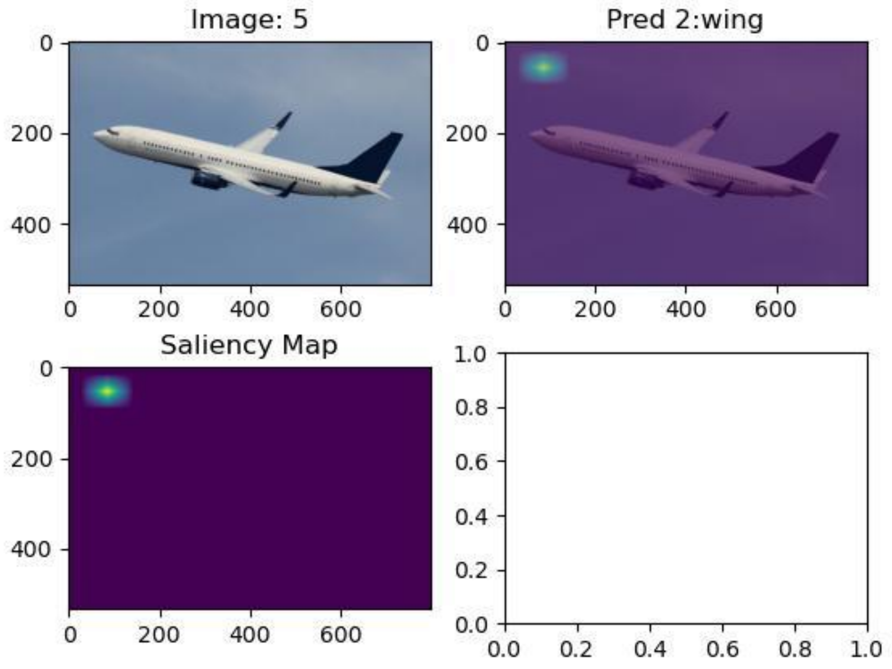
Ryan Filgas
Program 4
Computer Vision





Ryan Filgas
Program 4
Computer Vision





Ryan Filgas
Program 4
Computer Vision

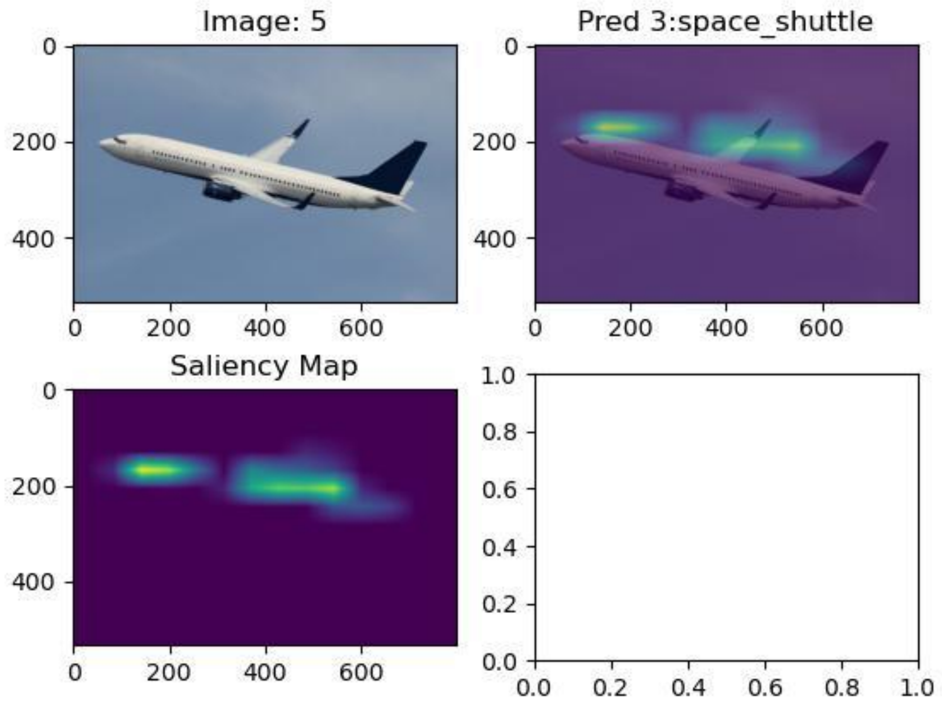


Image: 5
Pred 3:space_shuttle
Saliency Map