ELEC 475 Lab 5
Pet Nose Localization

Mark Benhamu 20173084, Ilir Gusija 20158525

Introduction

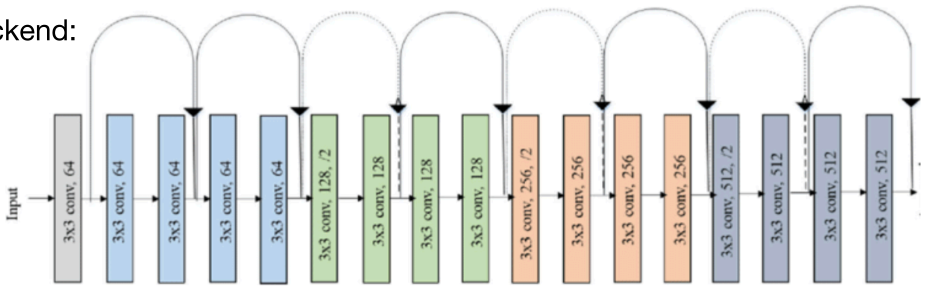
This project focuses on the task of key-point localization, specifically identifying the location of pet noses in images. The ability to accurately localize specific features in images has wide applications in areas such as facial recognition, augmented reality, and animal tracking. The project aims to develop a model capable of pinpointing the nose position of pets in various images.

Model Selection and Design

Backend

The ResNet18 model was chosen as the backend for this task due to its proven capability in feature extraction, particularly in deeper layers where complex features like eyes and noses are identified. ResNet18 offers a balance between depth and computational efficiency, making it suitable for feature localization tasks. Seen in the following figure is the backend of the generated model. It is comprised of the ResNet-18 Conv2d feedforward layers. The average pooling, final FC layer, and the SoftMax activation layer at the end of the ResNet-18 model were discarded. Pre-Trained weights from the ImageNet dataset were used for the backend.

Model Backend:



Frontend Iterations

i. FC -> (x ,y) Coordinates

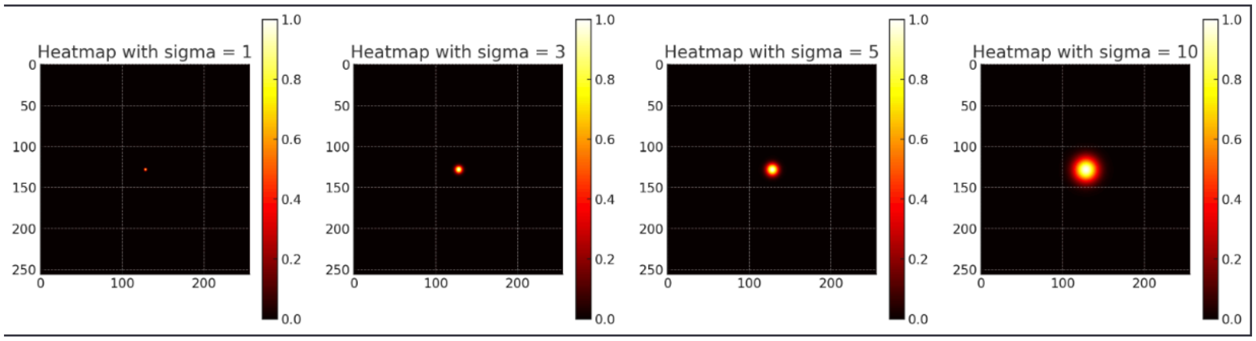
The initial model design output was a FC layer that took in the flattened output of the backend and outputted a 1-by-2 vector representing the (x, y) coordinate of the key-point. This approach faced challenges in training, with the loss remaining significantly high. This indicated that the model was struggling to directly regress to the precise coordinates of the nose.

i**i. Transition to Heat-map Approach**

To address the training difficulties, the model was adapted to a heat-map based approach. The first iteration involved a binary heat map target, represented as an NxN array filled with zeros at every index except that of the key-point which was assigned a value of one. This was done in order to visualize how the model was progressing during training and to increase the domain space to give the model more freedom than in the original setup. The training loss was very low from the start and showed minimal change over several epochs. This was attributed to the model outputting near-zero values across the heat-map, leading to a small loss due to the sparse nature of the binary target.

iii. Gaussian Heat-map

The final iteration of our model employs a Gaussian distribution centred around the nose key-point as the target for more precise localization. We experimented with various sigma values to fine-tune the spread of the heat-map. A radius that was too small resulted in inaccuracies, while one that was too large compromised the model's precision post-training. The following figure illustrates these examples.



For the front end, the model processes the output from the ResNet18 backbone through five stages of upsampling, which include:

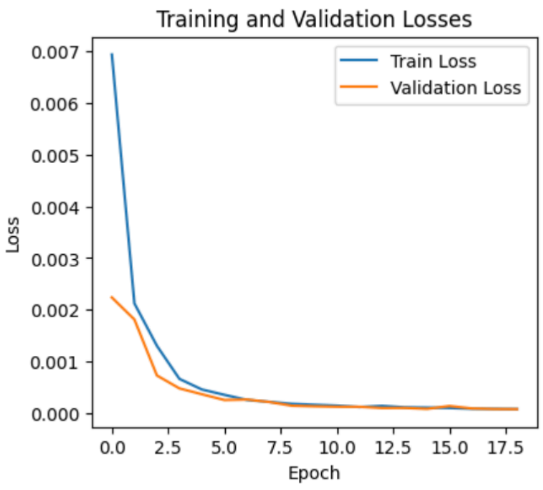
1. Bilinear Upsampling with a scale factor of 2 and aligned corners to increase the spatial resolution of the feature maps. To goal being to upsample back to the 256x256 image size so the heat map could be easily visualized on the image.
2. Conv2D Layers with a kernel size of 3 and padding of 1, designed to maintain the dimensions after upsampling while progressively reducing the depth from 512 to 1 across the sequences:
   - 512 → 256
   - 256 → 128
   - 128 → 64
   - 64 → 32
   - 32 → 1
2. Batch Normalization is applied after each Conv2D layer, with the depth size as its parameter, to stabilize and accelerate training.
3. Activation Functions are implemented as follows: ReLU for the initial four sequences to introduce non-linearity, and a Sigmoid activation in the final sequence to generate the output heat-map.

Training Details

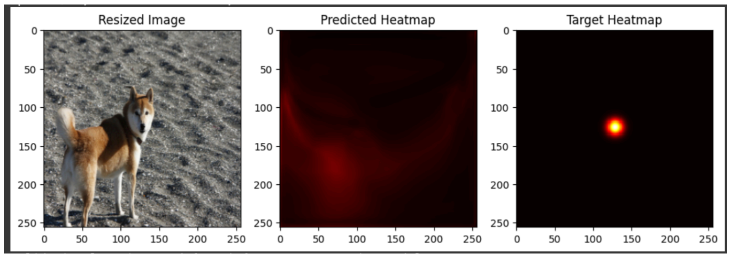
Hyperparameters

- Learning Rate: 0.001
- Batch Size: 32
- Number of Epochs: 30 (max)
- Early stopping patience: 3
- Target Size for Images: 256x256
- Sigma Value Used for Gaussian Target Generation: 7
- Loss Function: Mean Squared Error

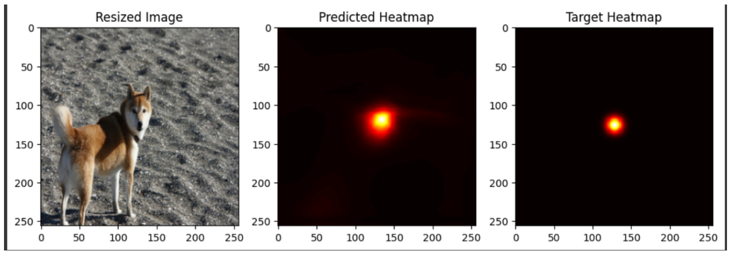
The model was trained on Colab Pro+ using a V100 GPU. The training process took approximately 30 minutes. Stopped at epoch in at epoch 17 after clear convergence. The loss plot is seen on the left, visualizations of the heat-map throughout the training process on the right.



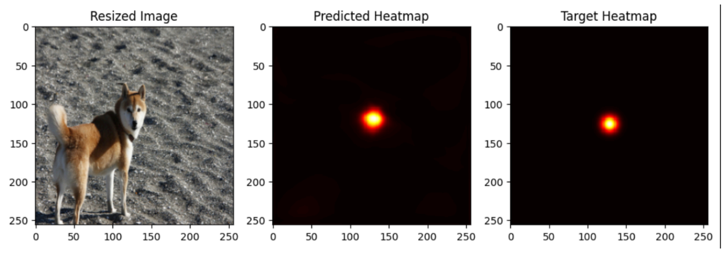
Epoch 1



Epoch 2

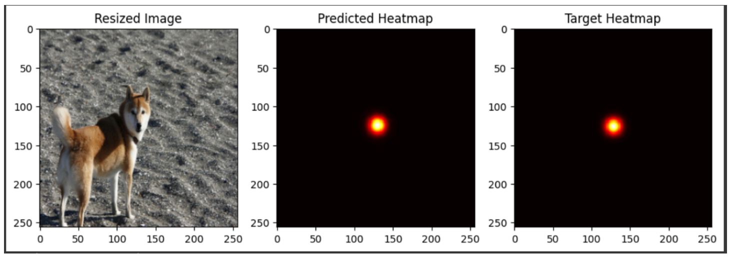


Epoch 3



⋮

Epoch 17

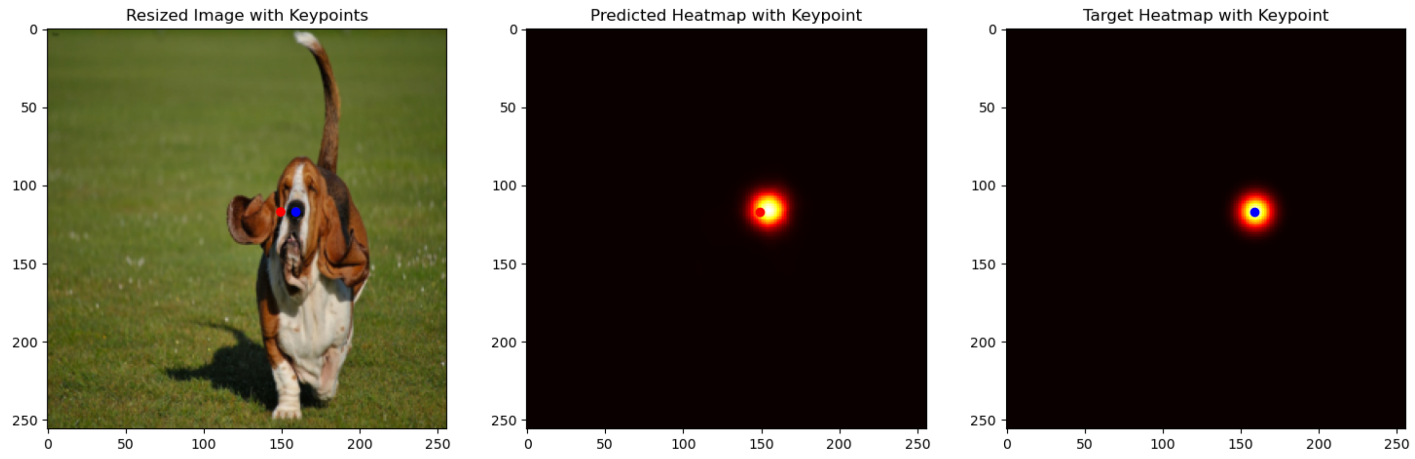


Localization Accuracy and Results

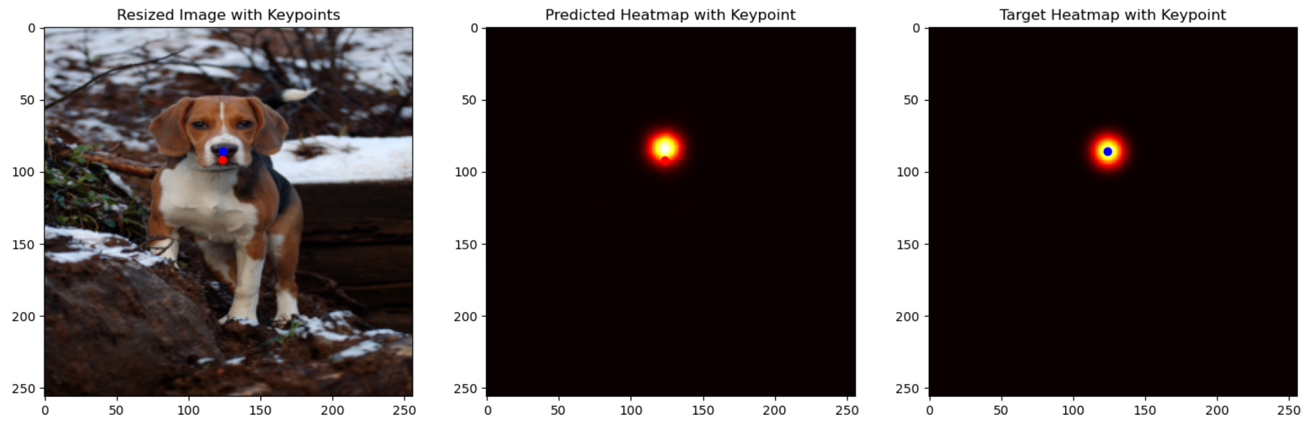
Accuracy Metrics

The localization accuracy was quantified using a mixture of PyTorch and Numpy library functions, full details can be found in the source code in test.py in the test function. The statistics were as follows:

- Minimum Distance: 0.4287
- Mean Distance: 12.35
- Maximum Distance: 123.35
- Standard Deviation: 12.86


Resized Image with Keypoints
Predicted Heatmap with Keypoint
Target Heatmap with Keypoint
0
50
100
150
200
250

Sample Result 1


Resized Image with Keypoints
Predicted Heatmap with Keypoint
Target Heatmap with Keypoint
0
50
100
150
200
250

Sample Result 2

Qualitative Results are very positive. The heat-map seems to be entered directly on the pet noses. Invariant to whether the pet is facing the camera or sideways. The centroid of each heat map is also represented overlayed over each image further showing its accuracy.

Inference Hardware and Performance
The inference tests were conducted on Lambda Labs PC with with two NVIDIA RTX A5500 GPUs with a total of 48 GB of VRAM, with the model processing the entire test set with batch size in 0.0 seconds (up to 15 decimal points, tracked using time.perf_counter()), i.e. it processes the images almost immediately.

Discussion

The performance of the system was above expectation. The project encountered several challenges, notably the inability to train for both of the initial model iterations. Achieving these accurate of results was doubtful at that time but through the model iterations and implemented changes great results were achieved.. Potential improvements would include reducing the size of the gaussian ring around the keypxoint while maintaining training accuracy. For images where the pet is further away, having the same size target ring makes it appear less accurate. The centre of these instances still appears to be very close to the key-points (x, y) coordinates. Further developments would be do go beyond a single key-point and localize multiple pet features, such as eyes, paws, ears, etc. In all the lab was very rewarding and the team is content with model performance.