Deep Semantic Hashing for Aerial Livestock Detection

Shosei Anegawa, Franz Kurfess, Sumona Mukhopadhyay

CAL POLY College of Engineering

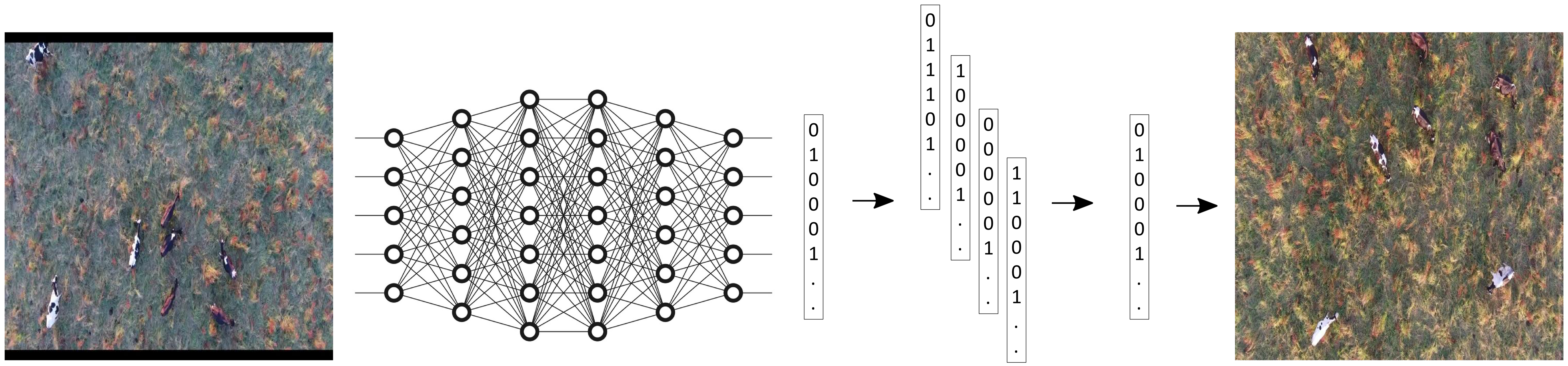

Figure 1) Flowchart for classification of images using hashing

Image is passed through neural network(AlexNet) to create 48-digit long binary hash code.

Database of previously transformed images is searched for similar codes

The model is trained to make similar codes for images with similar content, so we conclude that the original image must also have 8 cows

Figure 2) Faster R-CNN Object Detection Benchmark - Model is very accurate and gives exact location of sheep/cattle but is much slower and thus costs more power than hashing.

Introduction

The goal of this project is to be able to accurately detect and count livestock in footage captured by a drone in real time. The main problems with this arise from the fact that a drone can only carry limited computing resources, and hashing is conventionally thought of as a great method of doing image classification very quickly and thus even on low-power devices. In this project, we use both a Faster-RCNN, which is a state-of-the art object detection model (Figure 2) as a benchmark to develop a hashing model (Figure 1) that can perform a similar task much more quickly. These two models provide a trade-off between accuracy and speed, where the Faster-RCNN is more accurate and gives precise locations of the livestock in the image, while the hashing is 20-25x faster but is less accurate and only provides the number of livestock in the image. Given that the dataset is very limited in quantity, we also build a generative network to create more images for the model to train on so that it has a more diverse set of hash codes to reference.

Methods - Hashing

The objective of a hashing model is to produce a binary code – in our case 48 digits that are either -1 or 1. The model is then trained to make similar codes for similar images, and dissimilar codes for dissimilar images, and the codes are all stored for later use. Then, when the model encounters an image that it has never seen before, it compares the hash code derived from that image with those in the model's database. The model then concludes that the image with the most similar code must be of the same class as the original. In our case, we train the model to have similar hash codes if there are the same number of animals in the images.

Changes to loss function

Most hashing models employ a triplet loss function, where on each image they train to become more like a similar image, and less like a dissimilar image. Our best performing hashing model was the Deep Supervised Hashing algorithm, which uses a unique loss function to maximize discriminability of images. We apply one major change to the loss that has increased performance of the model, which is to dynamically weight the similarity or dissimilarity of images based on the difference in number of cows. This allows the model to generalize to new images better but is only useful for specific problems that involve counting such as this one.

Figure 3) Example GAN-generated image. While Image quality is poor, it allows for more diverse hash codes to be store so that the model can be more robust towards new data.

GAN for Image Synthesis

In order to counteract the greatest weakness of hashing, the lack of codes in our database to compare to, we also employ a Generative Adversarial Network to generate additional similar images to fill in more hash codes. The GAN can create images that are like the real images in the dataset but that may have animals in slightly different positions or have an additional animal somewhere in the image, at the cost of the images being either less realistic or lower quality. However, even with lots of noise in the images the model can generalize sufficiently well due to its convolutional structure, and these GAN-generated images are selected as "most similar" almost as often as real ones.

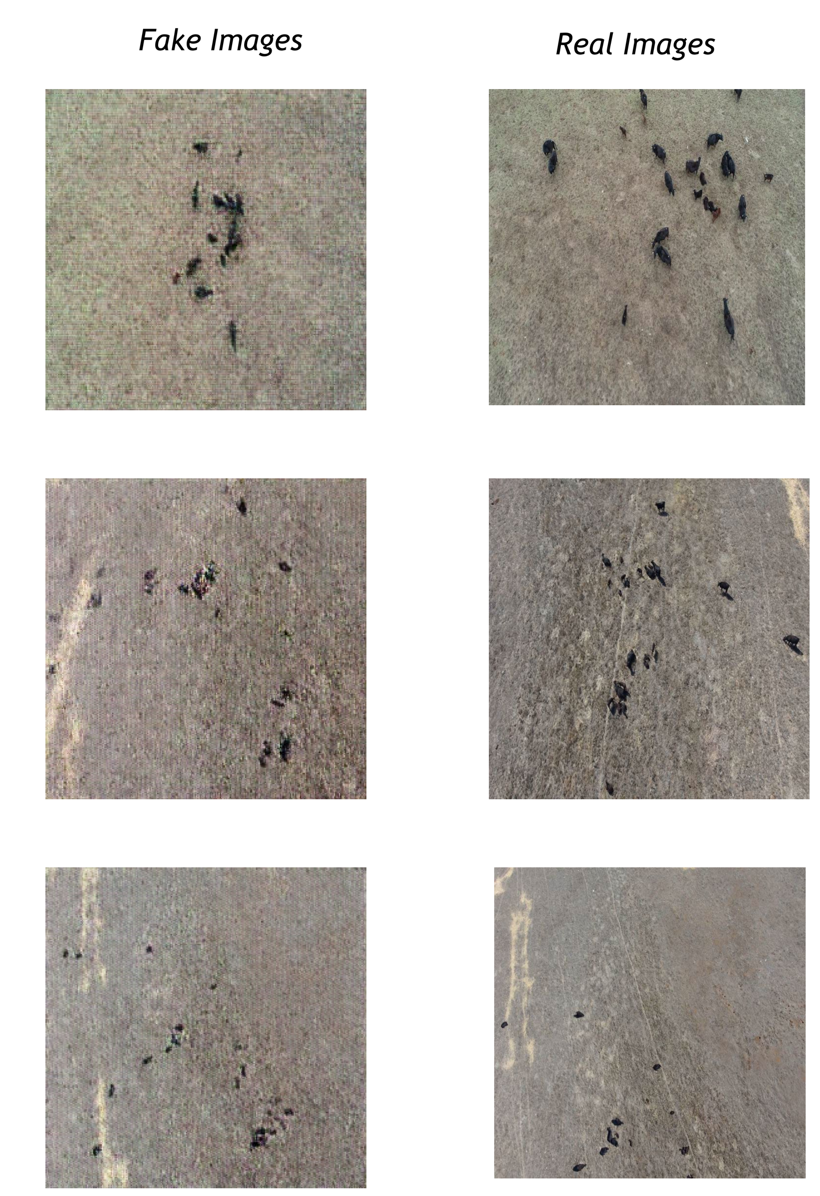

Figure 4) Generated GAN Images (Left) and similar real images (Right)