The class that the SVM predicts is the one with the highest score. The optimization method is Stochastic Gradient Descent (SGD) using mini batches of data.

4.2. Multiclass SVM + HOG + HSV

This multi class SVM+HOG+HSV is very similar as the vanilla version described above, except that it takes in HOG and HSV features of the image instead of raw image data. Conceptually, HOG captures texture but no color information from the image, while HSV captures color but no texture information from the image. By extracting these features independently and then concatenating during training time, we obtain a richer feature landscape.

4.3. 3-Layer ConvNet

The architecture of the 3-layer ConvNet consists of three sections. The first section consists of a convolutional layer, followed by a ReLU activation, and ending with max-pooling. The second section is the same as the first. The third section consists of a fully-connected layer, followed by a ReLU activation, and ending with a linear affine. The convolution layers use 32 5x5 filters with a stride of 1. Max-pooling is 2x2 (which essentially halves the planar dimensions) with a stride of 2. The fully-connected layer has 1024 nodes. For training, SGD with Adam optimization is used, and dropout is used for regularization [5],[6].

Adam is the state-of-the-art gradient update rule for ConvNets. It combines elements from RMSProp and momentum update. Dropout is a regularization technique that helps prevent ConvNets from overfitting. The idea that during each a training step, a random group of neurons are disabled, which helps prevent neurons from co-adapting (i.e. developing an overly strong dependence on one another). The 3-layer ConvNet takes in a raw image as a 150x150x3 dimensional array and classifies the input image to one particular tag.

4.4. 5-Layer ConvNet

The architecture for our 5-layer ConvNet is the similar to the 3-layer ConvNet, except there are two more [conv - relu - pool] layers appended. The parameters for the convolutional layer, max-pooling, and fully-connected layer are the same, and SGD with Adam optimization and dropout are used as well[6].

4.5. AlexNet

We use AlexNet as presented in this paper [7]. A unique feature of AlexNet is that is uses local response normalization to normalize the "brightness" of neurons. The local response normalization use the following equation to normalize the brightness of the neurons:

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^\beta$$

Where $a^i_{x,y}$ denotes the activity of a neuron computed by applying kernel i at position (x, y) [7].

The AlexNet consists of four sections. The first, second, and third sections consist of a convolutional layer, followed by a ReLU activation, max-pooling, and ending with local response normalization. The fourth section consists of a fully-connected layer, followed by a ReLU activation, and ending with a linear affine to obtain the class scores.

Max-pooling is 2x2 with stride 2 throughout the AlexNet (which essentially halves the planar dimensions at each step). The first convolutional layer has 64 filters, the second 128 filters, and the third 256 filters, where the filters are of size 3x3 with stride 1. The fully-connected layer has 1024 nodes. For training, SGD+Adam optimization is used along with dropout.

4.6. VGGNet

VGGNet uses very small convolution filters (3x3), which allows the depth to be increased with less overhead than if it used larger filters [8]. The VGGNet consists of 8 sections. The first 5 sections consist of two pairs of convolution layers and a ReLU activation, which are followed by max-pooling. The last 3 sections consist of fully-connected layers.

Max-pooling is 4x4 with a stride of 4 in the first section and 2x2 with a stride of 2 in the other sections. The convolution layer has 3x3 filters throughout the VGGNet, while the number of filters varies per section. The number of filters are 64, 128, 256, 512, and 512 for the first five sections respectively. The number of neurons in the fully-connected layers are 4096, 10, and 4 respectively for the last three sections. For training, SGD+Adam optimization is used along with dropout.

5. Results and Discussion

We experimented with methods like SVM, SVM+HOG+HSV, 3 and 5 layer Convolutional Neural Network, AlexNet and VGGNet. One of us also manually performed the task to get a measure of human performance for this task. The person trained on around 400 images and predicted the tags (i.e. morning, afternoon, evening, night) for around 200 images. Several aspects of the data make this task very difficult, which was made obvious when the person scored a test accuracy of around 40%.

We implemented the convolutional neural networks using the recently-released framework TensorFlow [9]. We used a keep-rate of 75% for dropout, and we used learning