EmberScan

John Mansfield

Figure 1: EmberScan Logo

EmberScan's Mission

EmberScan is developed with the idea of using A.I image processing to identify housefires using the Alexnet.

Every Year 358,300 homes have uncontrolled fires. EmberScan is a company dedicated to utilizing AI imaging technology for the vital task of identifying house fires and preventing housefires. No system is without its flaws and devices such smoke detectors can fail due to things such as neglect of maintenance such as changing batteries, all the way to manufacturing issues. EmberScan's mission is to offer a fail-safe solution to this problem by enabling home security cameras, or housing development cameras to detect house fires and promptly alert the appropriate authorities. This innovative use of AI image processing not only enhances fire safety but also ensures a more reliable and proactive response to fires, potentially saving lives and property.

What EmberScan is trained to differentiate

Figure 2: House on fire

Figure 3: Normal House

Structure of an Alexnet

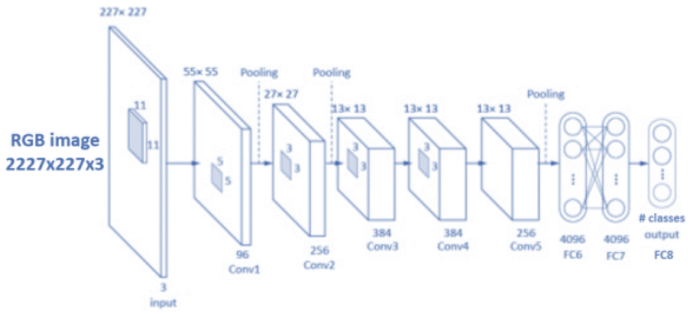


Figure 4: Alexnet

Alexnet's consist of an input layer which takes an RGB image input with the dimensions 227x227. Alexnets then have 5 convolutional layers which use filters to extract features from the image input. There are max pooling layers after the first and second convolutional layers which reduce the spatial dimensions of the feature maps while keeping the most important information. There are then 2 fully connected hidden layers which are a traditional neural network. The final layer is a third fully connected layer which is the softmax layer which produces a probability distribution of possible classifications...

What is EmberScan trained off of

EmberScan is trained off the following data deck House Fires Data Deck

Feature Maps

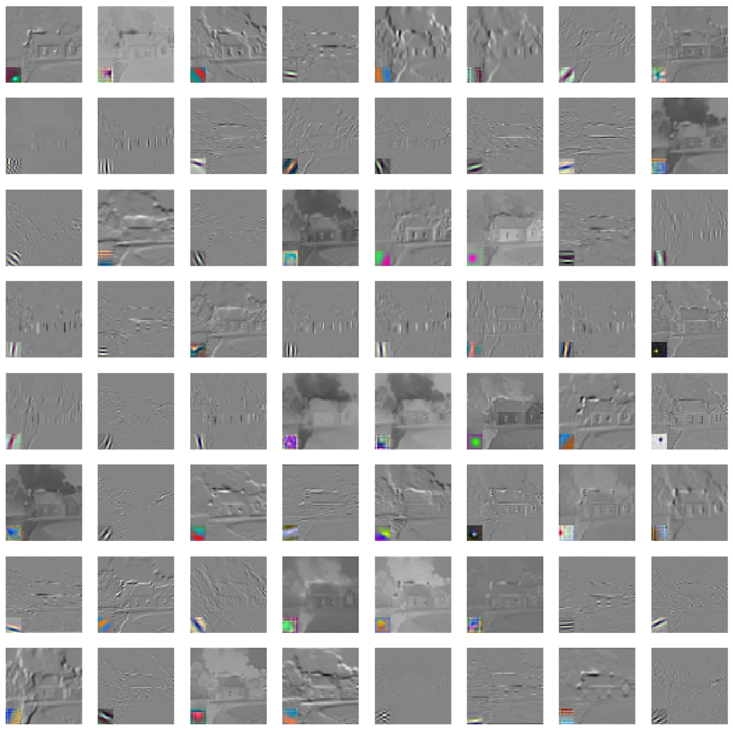


Figure 5: Filters and Feature Map

Training accuracy and loss

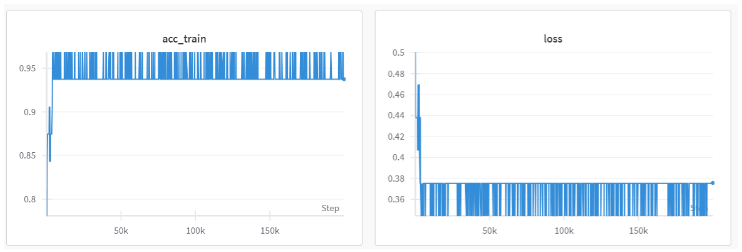


Figure 6: Training accuracy and loss

Working Model utilizing Alexnet and explanation of code

Working model

Code Overview

The provided code performs several tasks and utilizes various Python libraries to achieve its objectives. Here's a high-level overview of the code's functionality:

Importing Libraries

The code starts by importing necessary Python libraries, including wandb for experiment tracking, pdf2image for PDF to image conversion, and torch for deep learning functionalities. Additionally, it imports pre-trained models from torchvision.models and other utilities.

GPU Setup

The code checks for the availability of a GPU and assigns the computing device accordingly. This is important for leveraging GPU acceleration in deep learning tasks, which can significantly speed up model training.

Utility Functions

Several utility functions are defined to simplify various tasks within the code, such as data preprocessing, plotting, and random data generation. These functions enhance code readability and maintainability.

Loading a Pretrained AlexNet Model

A pretrained AlexNet model is loaded using the torchvision.models.alexnet function. This model is a deep convolutional neural network designed for image classification tasks and is known for its effectiveness in such applications.

Loading and Preprocessing Images

The code retrieves slides from a Google Slides URL and converts them into images. These images are then preprocessed and loaded into PyTorch tensors for use with the AlexNet model. This step prepares the data for classification.

Making Predictions

The pretrained AlexNet model is employed to make predictions (classifications) on the preprocessed images. The model's predictions are based on its understanding of the features present in the images.

Handling Predicted Labels

The code extracts the predicted class labels from the model's output and prints them. These labels are mapped to meaningful categories using the ImageNet labels provided in the code.

Creating Training Data

The code creates training data (X and Y) for a linear model. The data consists of 50 images divided into two classes, with 25 images in each class. One set is of houses on fire, and the other houses not on fire.

Defining Softmax and Cross-Entropy Functions

Functions for calculating softmax and cross-entropy loss are defined. These are fundamental components for training and evaluating classification models.

Generating Random Data

The code includes functions for generating random data, particularly truncated normal random numbers and a truncated normal distribution. These functions might be used for data augmentation or synthetic data generation.

Training a Linear Model

The code initializes hyperparameters and weight matrices for a linear model. It then uses the Adam optimizer to train the linear model for a specified number of epochs. During training, loss and accuracy metrics are logged and tracked using Weights and Biases (wandb).

Experiment Tracking with Weights and Biases (wandb)

Throughout the training process, the code uses Weights and Biases (wandb) to log and visualize training progress. This includes tracking loss and accuracy to monitor how the linear model is learning from the data.

Yolo Example image and code link

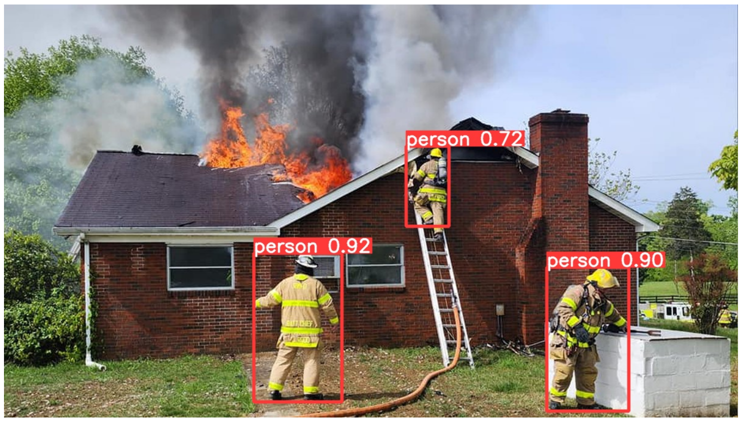
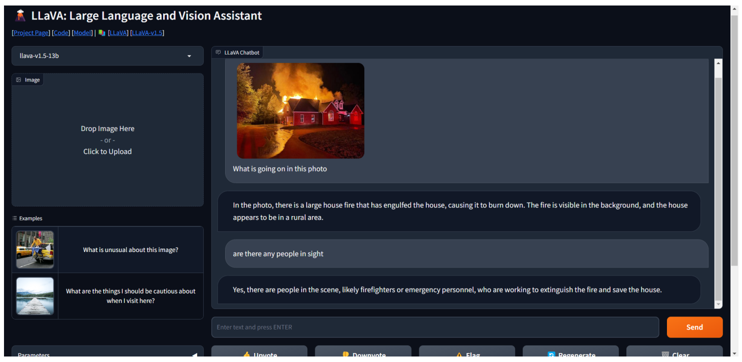


Figure 7: Yolo example

Yolo Model Link

LLava Test



LLaVA: Large Language and Vision Assistant
[Project Page] [Code] [Model] | [LLaVA] [LLaVA-v1.5]
Image
Drop Image Here
- or -
Click to Upload
Examples
LLaVA Chatbot
Enter text and press ENTER
Send
Parameters
Upvote
Downvote
Flag
Regenerate
Clear

llava-v1.5-13b

What is unusual about this image?

What are the things I should be cautious about when I visit here?

What is going on in this photo

In the photo, there is a large house fire that has engulfed the house, causing it to burn down. The fire is visible in the background, and the house appears to be in a rural area.

are there any people in sight

Yes, there are people in the scene, likely firefighters or emergency personnel, who are working to extinguish the fire and save the house.

Figure 8: LLava Test

It can be seen that LLava will make up information if it does not understand everything about the photo. Llava was however able to identify that the house was on fire.

Contact Information

Email: mansfieldj2019@fau.edu