**1. Overview of SSD**

The Single Shot MultiBox Detector (SSD) is a popular convolutional neural network (CNN) designed for object detection. SSD achieves high-speed and accurate detection by predicting object categories and bounding box offsets directly from feature maps in a single forward pass. Unlike region-based approaches, SSD eliminates the need for regional proposals, making it suitable for real-time applications. Key features include:

**2. Goals**

1. Implement a robust SSD model for real-time object detection.
2. Optimize the network for speed and accuracy on custom datasets.
3. Mitigate overfitting and ensure generalization across various environments.
4. Evaluate model performance using relevant metrics and visualizations.

**3. Comparison to YOLO**

SSD and YOLO (You Only Look Once) are both real-time object detection architectures, but they differ in key aspects:

* Prediction Mechanism:
  + SSD uses multiple feature maps for detection at various scales, improving performance on smaller objects.
  + YOLO predicts bounding boxes and class probabilities from a single feature map.
* Speed and Accuracy:
  + YOLO is faster but may sacrifice accuracy, especially for small objects.
  + SSD offers a better trade-off between speed and accuracy.
* Complexity:
  + SSD’s use of default boxes and multi-scale features adds complexity compared to YOLO’s simpler architecture.

**4. Overfitting**

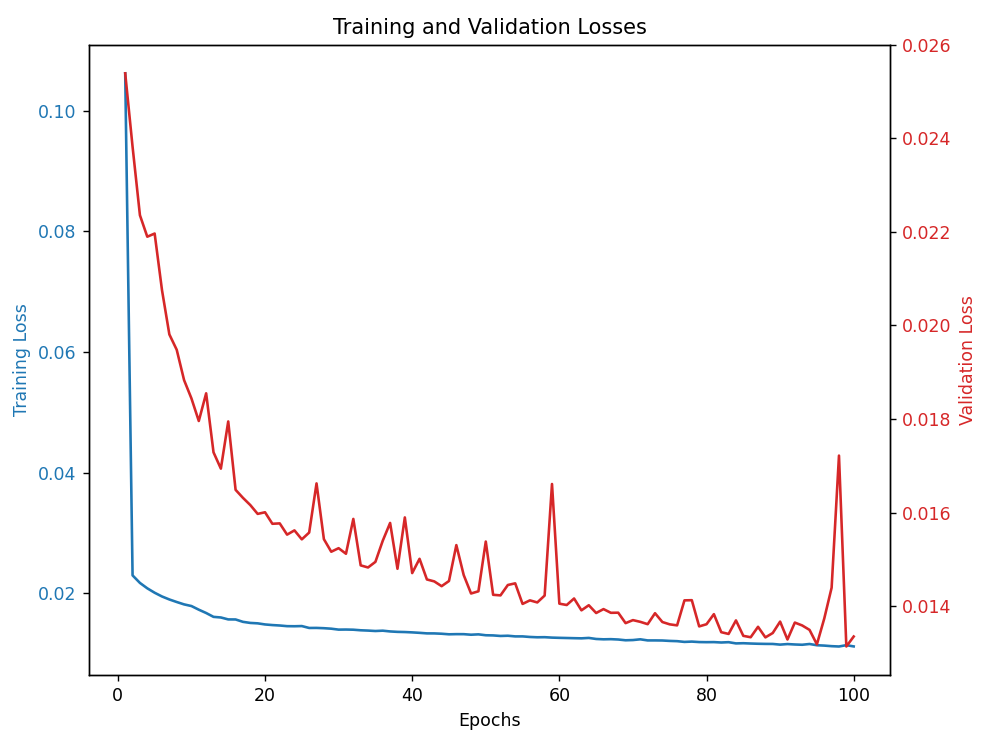
Overfitting occurs when the SSD model memorizes training data rather than generalizing. Strategies to address this include:

* **Regularization:** L1 loss and weight decay has been implemented in the code.
* **Early Stopping:** Overtraining a model has a tendency to cause overfitting. Early stopping detects when the learning of the model has halted and ends training early.,
* **Learning Rate Optimization:** By creating a scheduler, it can allow our code to learn slower as time passes, when learning fast is detrimental to our model.
* **Invalid Data Filtering:** While training, false data can cause the model to learn wrong. By filtering blatantly wrong data out we can cause our data to learn better. Negative Masking helps in this.

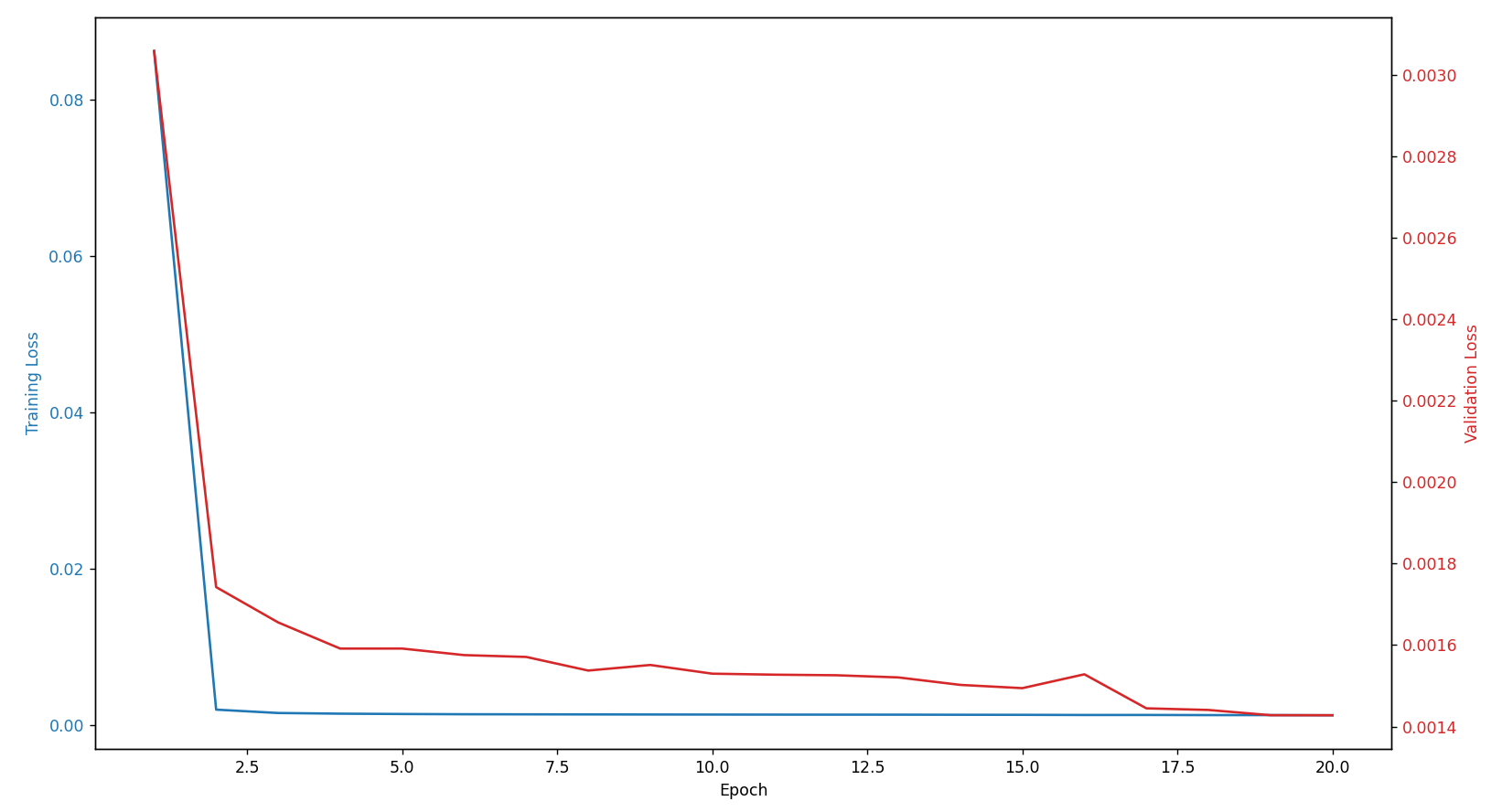
**5. Metrics and Graphs**

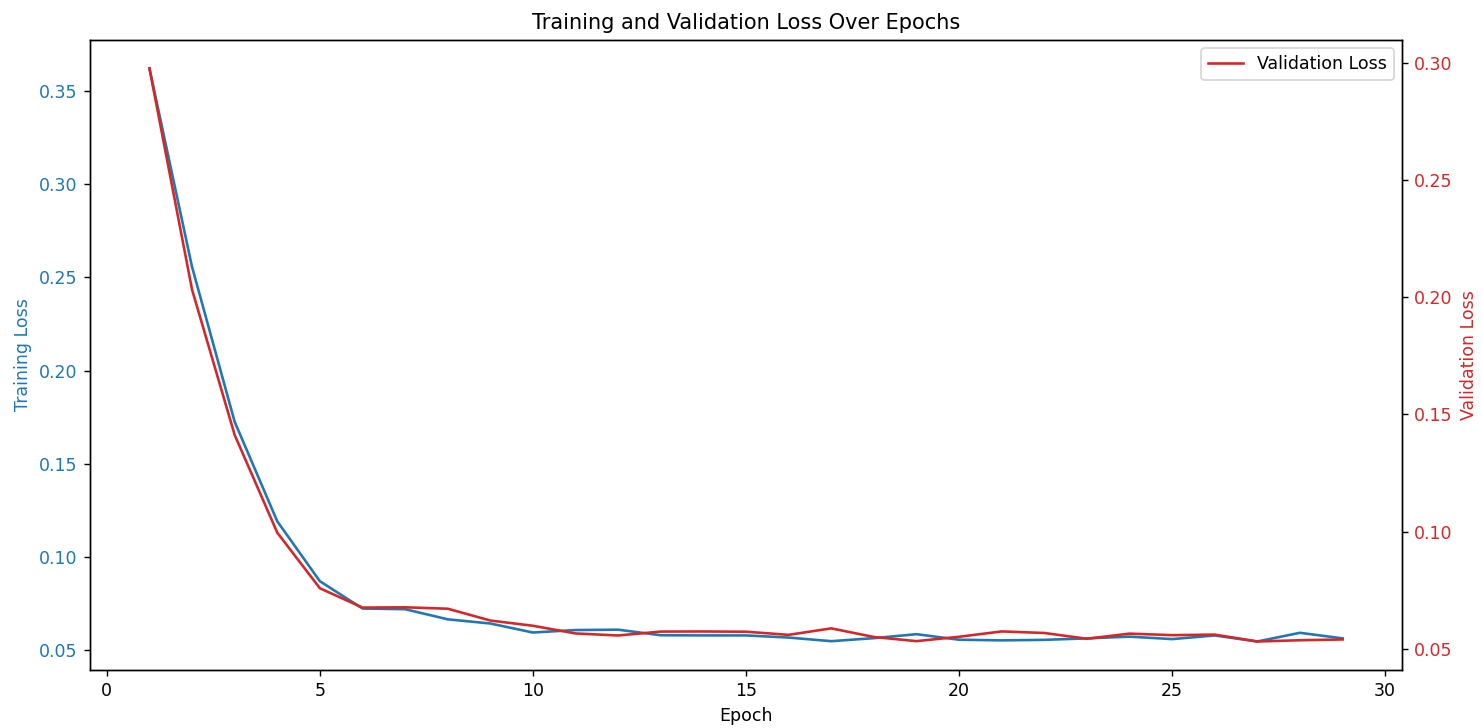
To evaluate SSD performance, key metrics include:

1. **Validation loss, Training loss and Time:** Time taken to process a single image, indicative of real-time capabilities.

At the start of the project overfitting was a very big issue. In this graph no overfitting prevention technique was used. This caused instability and overfitting to be prevalent

Some iterations later I added early stopping and L1 regularization. While the instability went away there was still a gap between training and validation loss.

****

Towards the end of the project I implemented schedulers, iou validation, negative masking and class weights. These techniques allowed my code to learn more smoothly and consistently. 

1. **Precision Recall F1:** Metrics that show how good our model is at giving reliable outputs:

**Metrics at the Metrics at the**

**beginning of the project end of the project**

A screen shot of a computer

Description automatically generated A screen shot of a computer

Description automatically generated

1. **IoU Thresholds:** Evaluates the localization accuracy of predicted bounding boxes. This is included in our evaluation function of the code. And adds its impact to validation loss.

**6. Conclusion**

The Single Shot MultiBox Detector (SSD) is a powerful framework for real-time object detection, balancing speed and accuracy through its multi-scale design. Addressing challenges such as overfitting and optimizing specific datasets ensures its effectiveness in practical applications.