# SiamFC Tracker
# Comparison from multiple perspectives

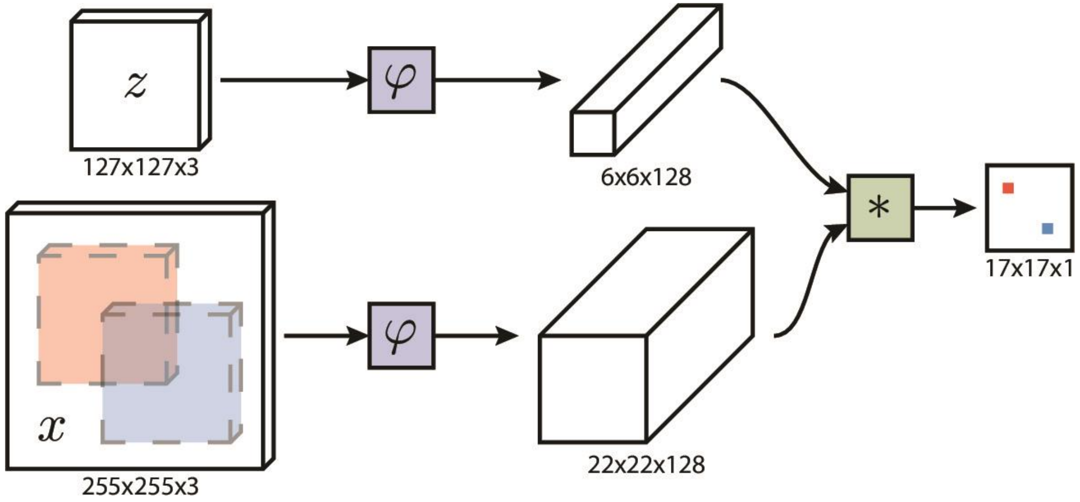
Kosuke Honda

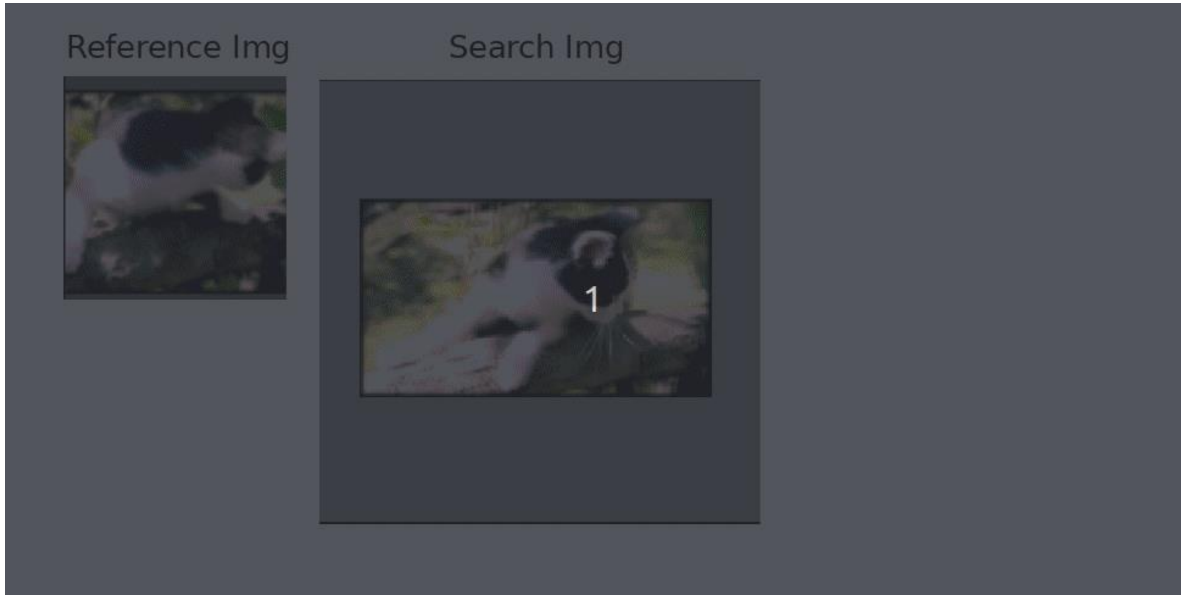
SiamFC Tracker

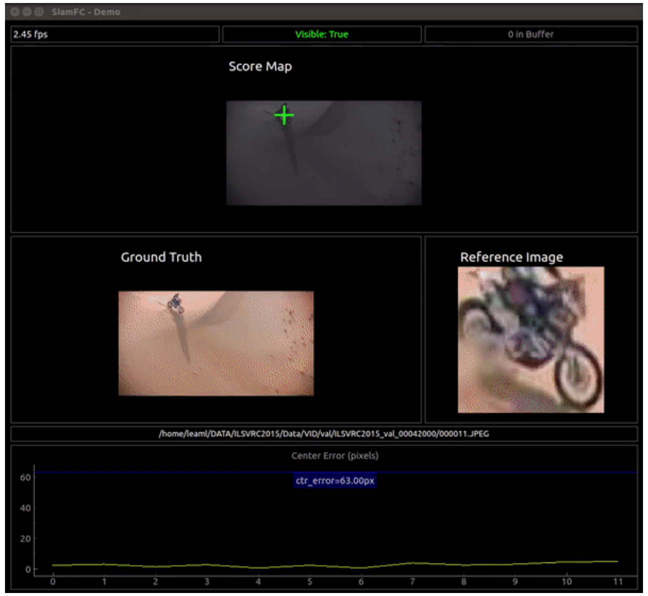
Comparison from multiple perspectives

1. Backbone network
2. Training dataset
3. Loss function
4. Optimizer

# SiamFC(Fully-Convolutional Siamese Networks for Object Tracking)



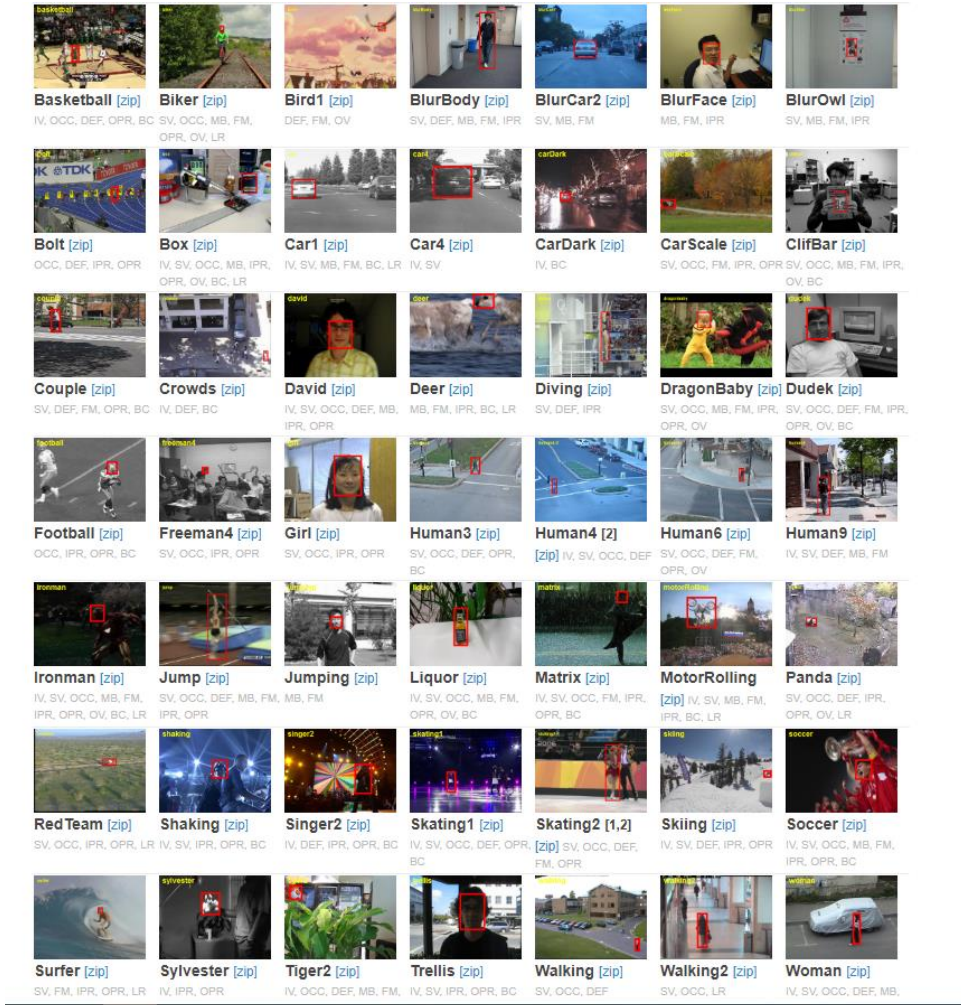



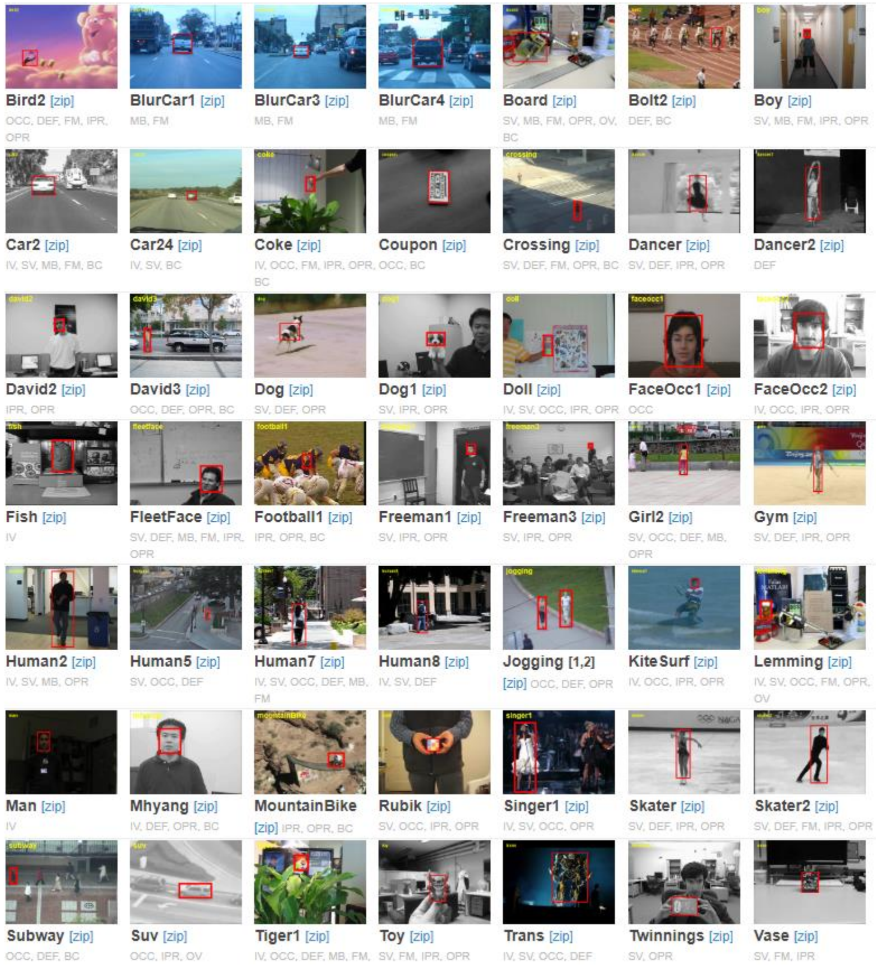



# Test Dataset:

OTB2015 : Object Tracking Benchmark(2015)

100 sequences

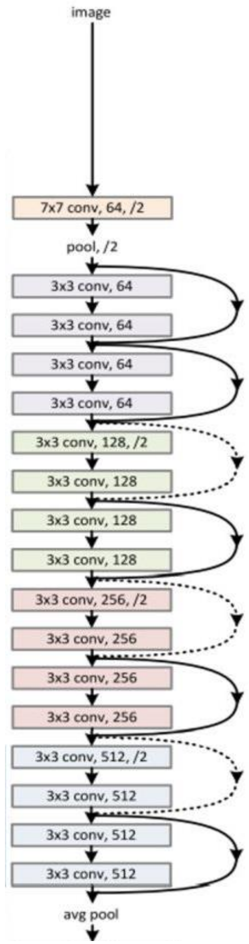


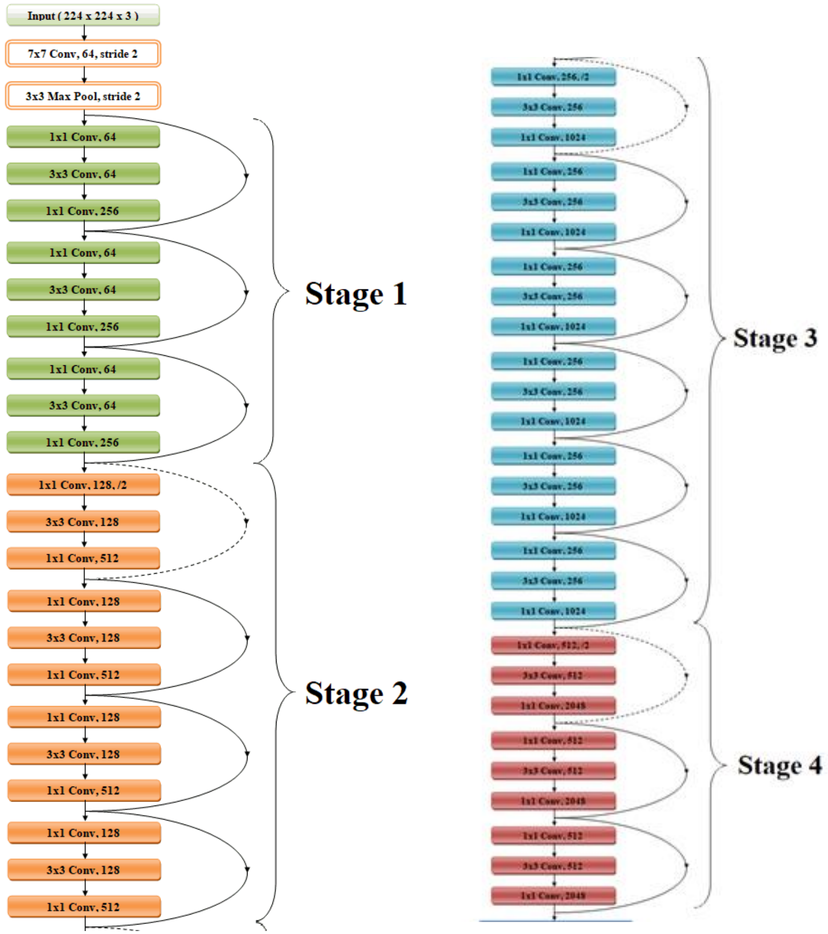
# Comparison Backbone Network

## Backbone network details

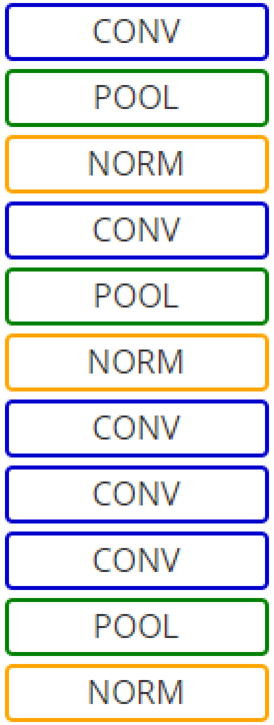
### Residual Network 18-layer




### Residual Network 50-layer

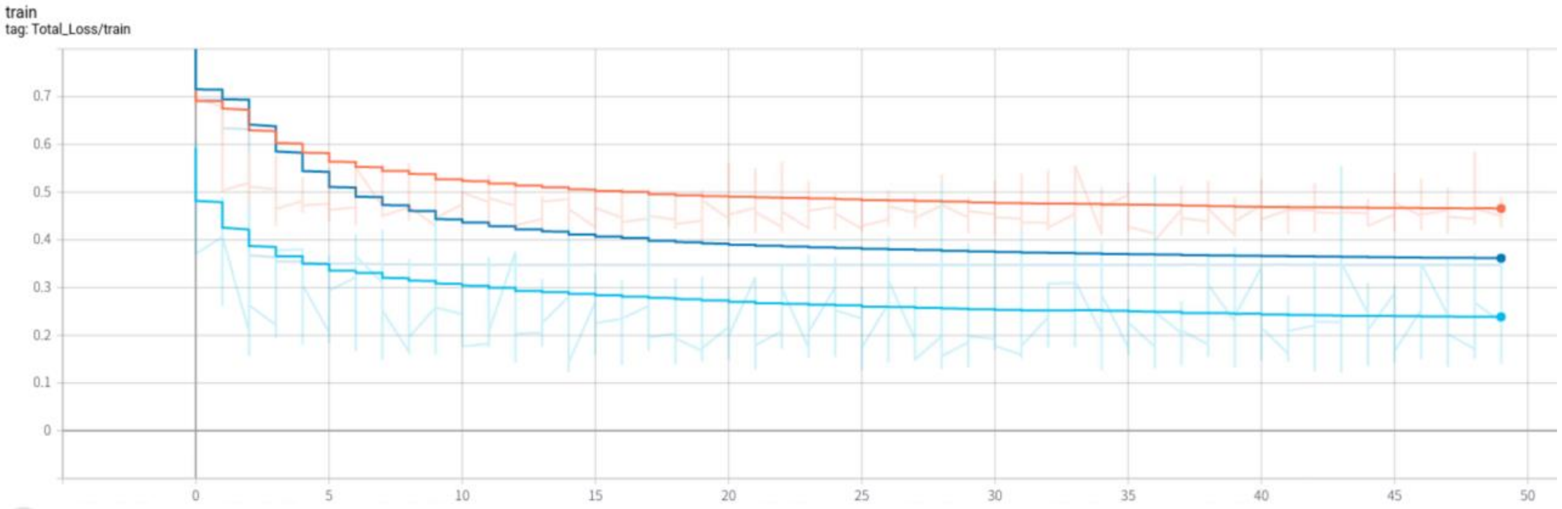



### Alex Network



# Comparison Backbone Network

## Learning Curve




Orange : ResidualNet-18
Blue : ResidualNet-50
Light Blue : AlexNet

## Performance

| | Res-18 | Res-50 | AlexNet |
|---|---|---|---|
| Precision | Cannot track due to lack of memory | Cannot track due to lack of memory | 0.788 |
| Success rate | Cannot track due to lack of memory | Cannot track due to lack of memory | 0.593 |

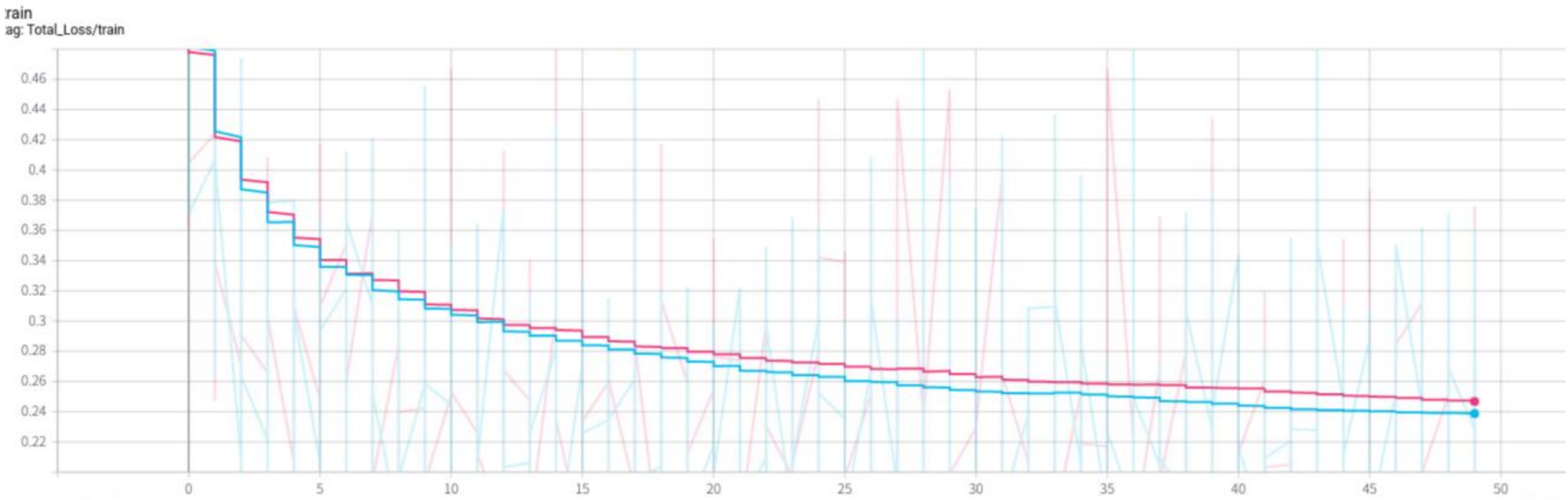
Training Dataset : Got-10k
Loss Function : Balanced Cross Entropy Loss
Optimizer : SGD-Momentum

**Success rate** :Average success rate for each threshold. Succeed if IoU is above the threshold. (threshold = 0.6)

**Precision** : If the distance between the prediction center coordinates and the ground truth center coordinates is within a threshold, it is considered successful.

# Comparison training datasets

## Learning Curve




Red : ImageNet-VID
Blue : Got-10k

## Performance

| | ImageNet-VID | Got-10k |
|---|---|---|
| Precision | 0.739 | 0.788 |
| Success rate | 0.547 | 0.593 |

Backbone Network (feature detection) : AlexNet
Loss Function : Balanced Cross Entropy Loss
Optimizer : SGD-Momentum

**Success rate** :Average success rate for each threshold. Succeed if IoU is above the threshold. (threshold = 0.6)

**Precision** : If the distance between the prediction center coordinates and the ground truth center coordinates is within a threshold, it is considered successful.

# Comparison LossFunction

## Loss Function details

Binary Cross Entropy Loss

$$\mathrm{CE}(p_t) = -\log(p_t).$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise.} \end{cases}$$

Focal Loss

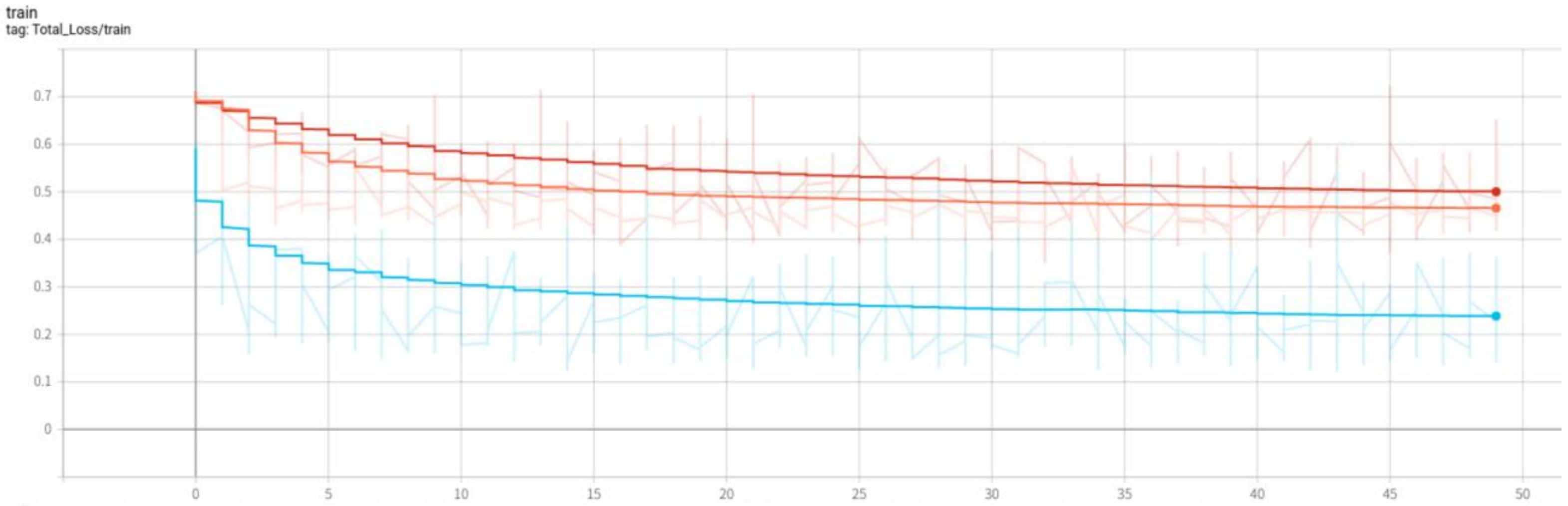
$$\mathrm{FL}(p_t) = -(1 - p_t)^{\gamma}\log(p_t).$$

Balanced Cross Entropy Loss

$$\mathrm{CB}_{\mathrm{sigmoid}}(\mathbf{z}, y) = -\frac{1-\beta}{1-\beta^{n_y}} \sum_{i=1}^{C} \log\left(\frac{1}{1+\exp(-z_i^t)}\right)$$

# Comparison LossFunction

## Learning Curve




Red : Cross Entropy Loss
Orange : Focal Loss
Light Blue : Balanced Cross Entropy Loss

## Performance

| | CrossEntropy | Focal | BalancedCE |
|---|---|---|---|
| Precision | 0.691 | 0.756 | 0.788 |
| Success rate | 0.511 | 0.569 | 0.593 |

Backbone Network (feature detection) : AlexNet
Training Dataset : Got-10k
Optimizer : SGD-Momentum

**Success rate** :Average success rate for each threshold. Succeed if IoU is above the threshold. (threshold = 0.6)

**Precision** : If the distance between the prediction center coordinates and the ground truth center coordinates is within a threshold, it is considered successful.

# Comparison Optimizer

## Optimizer

1. SGD（Stochastic Gradient Decent：確率的勾配降下法）

$$W \leftarrow W - \eta \frac{\partial L}{\partial W}$$

（W：パラメータ、η：学習率、L：損失関数、dL / dW：勾配）

2. Momentum

$$v \leftarrow \alpha v - \eta \frac{\partial L}{\partial W}$$

$$W \leftarrow W + v$$

3. Adam（Adaptive Moment Estimation）

$$m \leftarrow \beta_1 m + (1 - \beta_1) \frac{\partial L}{\partial W}$$

$$v \leftarrow \beta_2 v + (1 - \beta_2) \left( \frac{\partial L}{\partial W} \right)^2$$

$$\hat{v} = \frac{v}{1 - \beta_2}$$

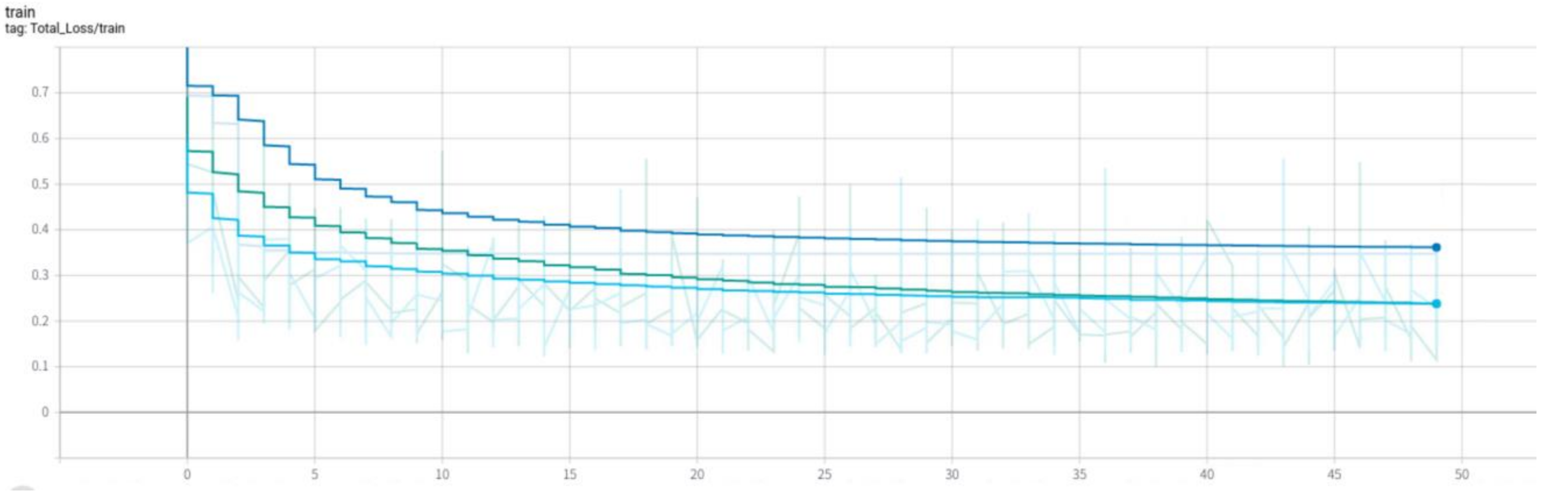
$$\hat{m} = \frac{m}{1 - \beta_1}$$

$$W \leftarrow W - \frac{\eta \hat{m}}{\sqrt{\hat{v}} + \epsilon}$$

```
torch.optim.Adam(params, lr=0.001, betas=(0.9, 0.999), eps=1e-08, weight_decay=0, amsgrad=False)
```

# Comparison Optimizer

## Learning Curve




Blue: SGD(No momentum)
Green : Adam
Light Blue : SGD(Momentum)

## Performance

| | SGD | Adam | Momentum |
|---|---|---|---|
| Precision | 0.732 | 0.773 | 0.788 |
| Success rate | 0.543 | 0.575 | 0.593 |

Backbone Network (feature detection) : AlexNet
Training Dataset : Got-10k
Loss function : Balanced CE loss

**Success rate** :Average success rate for each threshold. Succeed if IoU is above the threshold. (threshold = 0.6)

**Precision** : If the distance between the prediction center coordinates and the ground truth center coordinates is within a threshold, it is considered successful.

# Conclusion and Discussion

## Backbone network :

・"SiamFC" is a structure that trains a similarity score map between the target image and the search image by convolution it with the target image against the search image.

So using a **deep layered network like ResNet** as a Backbone is not a good match due to the **increased number of dimensions and weight parameters**.

## Loss function :

・Class Balanced Cross entropy loss gave the best results.
There was also an example of a combination of Focal loss and Class Balance loss, which I would like to try.

## Optimizer :

・The results using **Momentum as an optimizer were the best**, but we believe that adjusting Adam's hyperparameters would give better results. However, it is difficult to optimize Adam's hyperparameters because of the long training time per session.