

# Num of parameters

- Seperable conv architecture : 12,823,616
- MC3 architecture : 89,887,488
- MC4 architecture : 54,790,400
- MC5 architecture: 27,656,448
- Author's architecture : 27,655,936

## Original c3d architecture

```
self.conv4a = nn.Conv3d(256, 512, kernel_size=(3, 3, 3), padding=(1, 1, 1))
self.conv4b = nn.Conv3d(512, 512, kernel_size=(3, 3, 3), padding=(1, 1, 1))
self.pool4 = nn.MaxPool3d(kernel_size=(2, 2, 2), stride=(2, 2, 2))

self.conv5a = nn.Conv3d(512, 512, kernel_size=(3, 3, 3), padding=(1, 1, 1))
self.conv5b = nn.Conv3d(512, 512, kernel_size=(3, 3, 3), padding=(1, 1, 1))
self.pool5 = nn.MaxPool3d(kernel_size=(2, 2, 2), stride=(2, 2, 2), padding=(0, 1, 1))

self.relu = nn.ReLU()
```

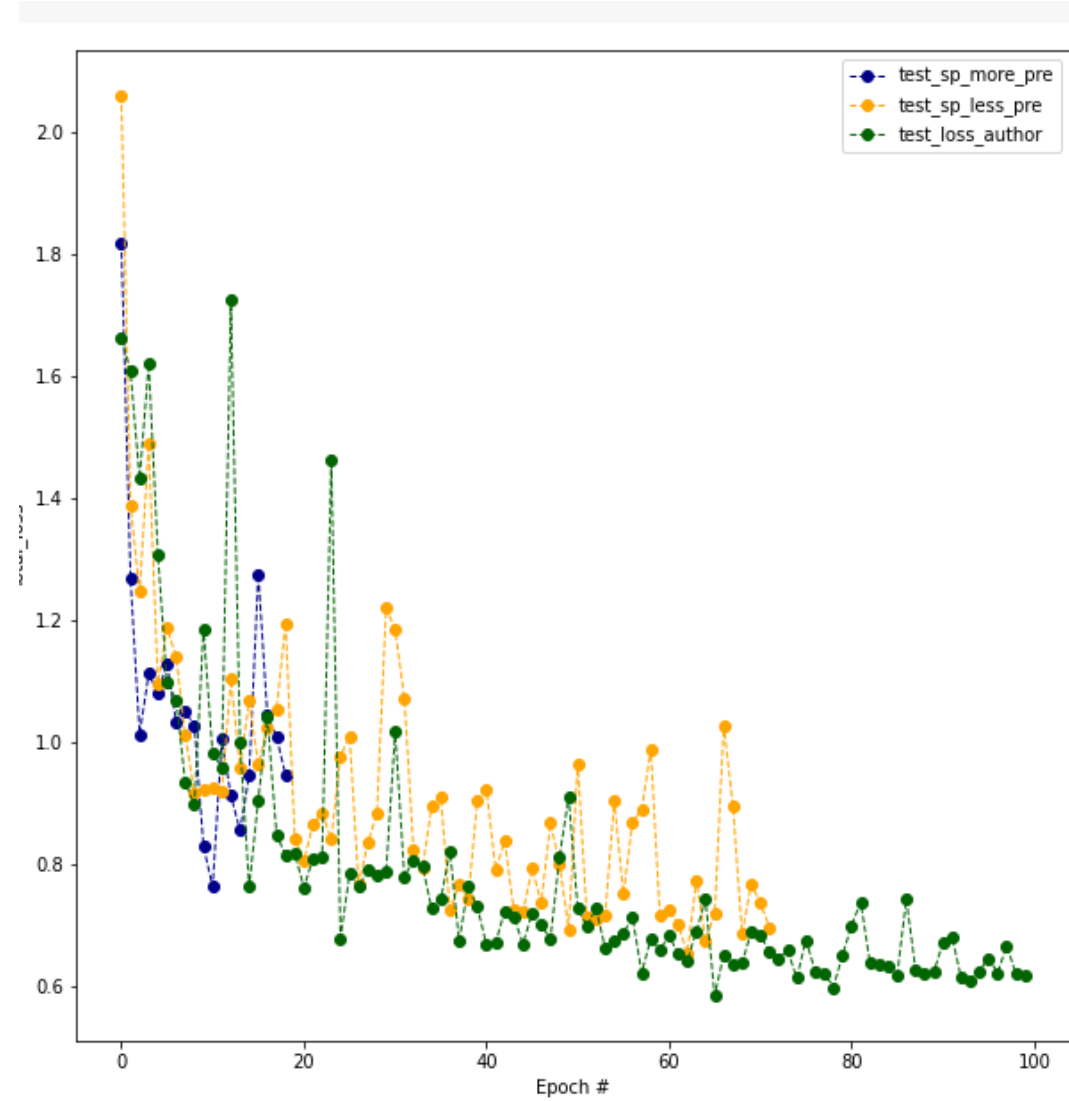
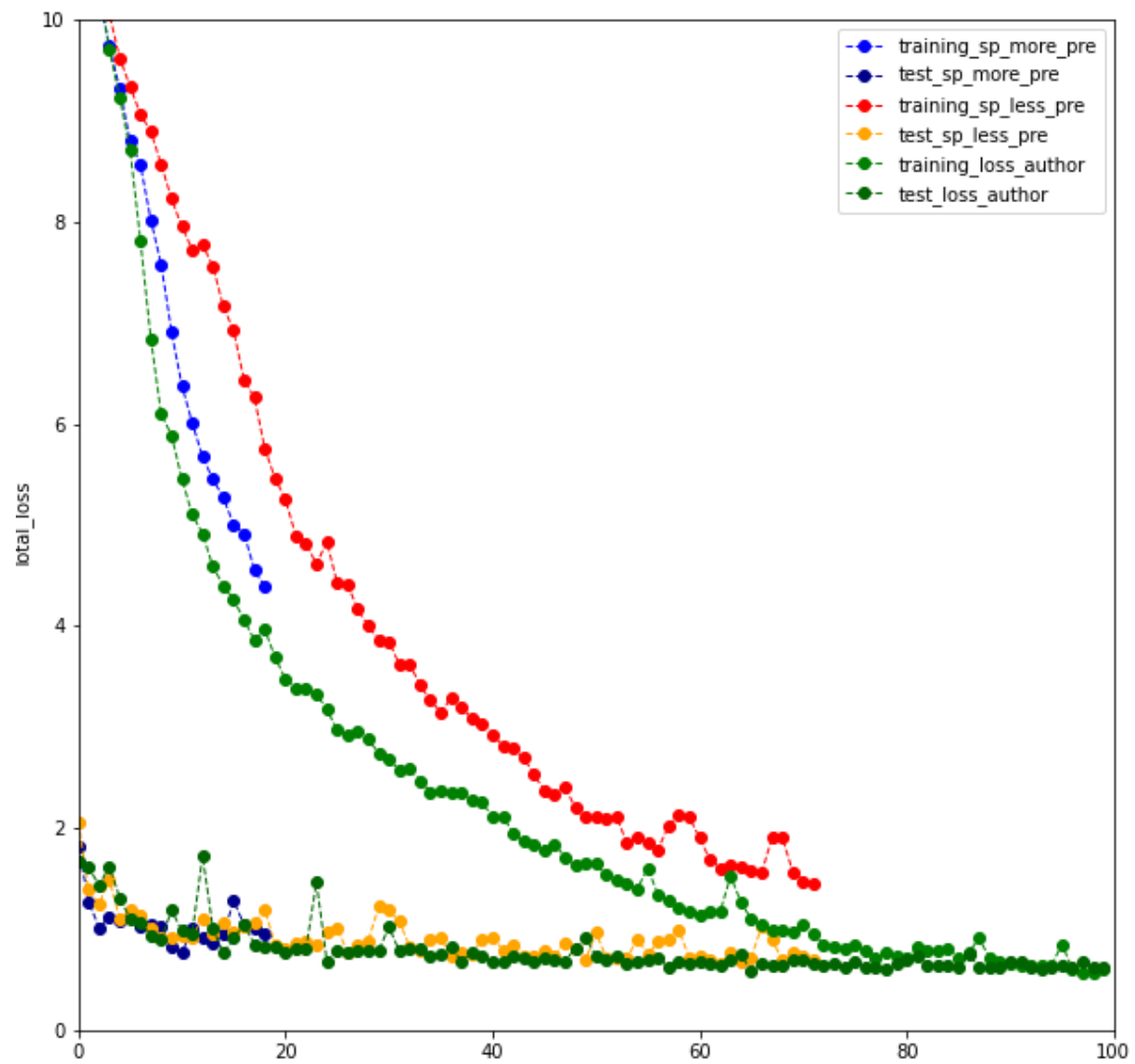
## Changed Architecture (MC4)

```
self.conv4a = nn.Conv2d(1024, 2048, kernel_size=(3, 3), padding=(1, 1))
self.conv4b = nn.Conv2d(2048, 1024, kernel_size=(3, 3), padding=(1, 1))
self.pool4 = nn.MaxPool2d(kernel_size=(2, 2), stride=(2, 2))

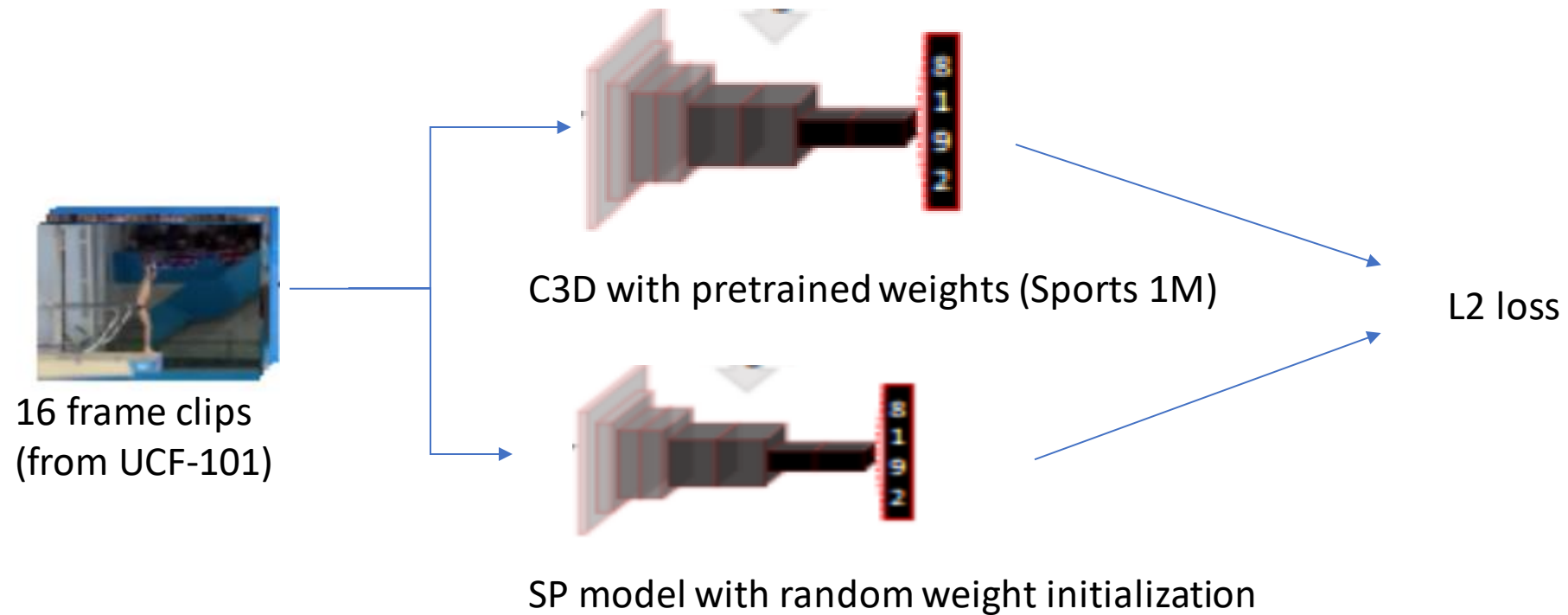
self.conv5a = nn.Conv2d(1024, 1024, kernel_size=(3, 3), padding=(1, 1))
self.conv5b = nn.Conv2d(1024, 512, kernel_size=(3, 3), padding=(1, 1))

self.pool5 = nn.MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=(1, 1))

self.relu = nn.ReLU()
```



# Pre training method



# Knowledge Distillation

- While looking for better approaches to approximate the weights from the pre trained model, we found out about knowledge distillation
- "DeepVID: Deep Visual Interpretation and Diagnosis for Image Classifiers via Knowledge Distillation" -Junpeng Wang, Liang Gou, Wei Zhang, Hao Yang, and Han-Wei Shen, Member, IEEE
- "Distilling the Knowledge in a Neural Network" - Hinton et. Al.

