Video Action Recognition

Ekapol Chuangsuwanich
Nvidia IVA workshop

CHULA ENGINEERING
Foundation toward Innovation

COMPUTER

Action Recognition in Videos

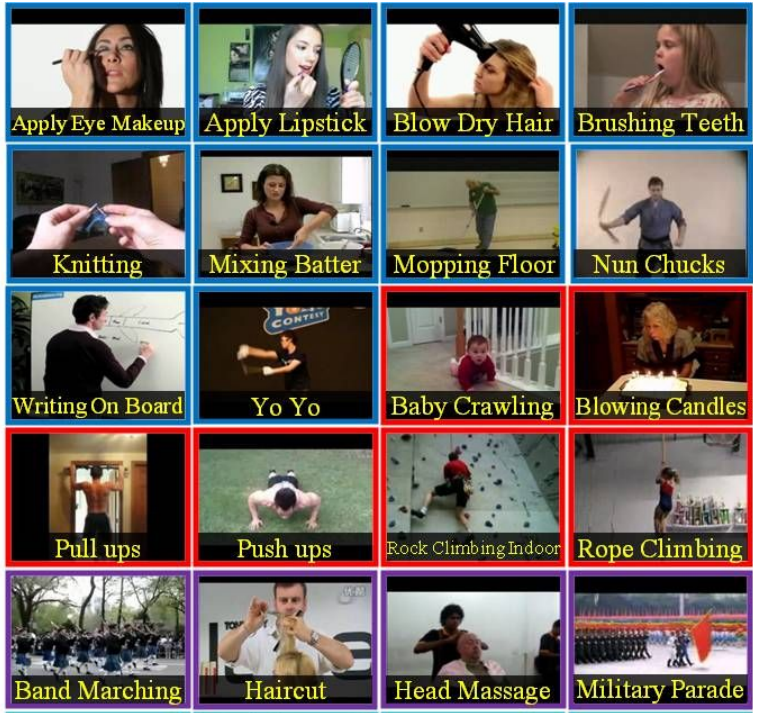
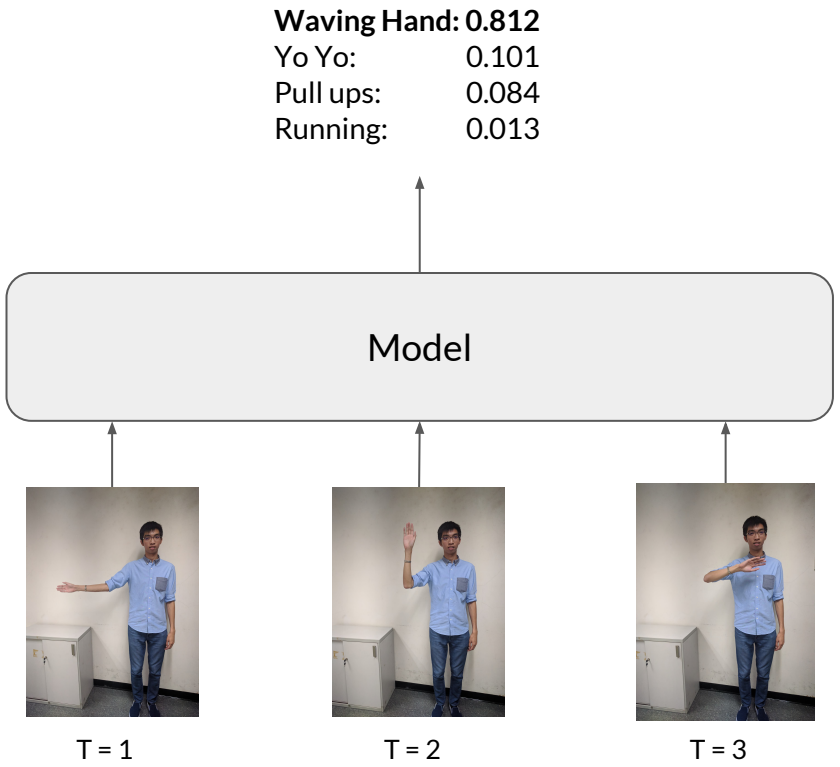


Image: UCF101



Action Recognition in Videos

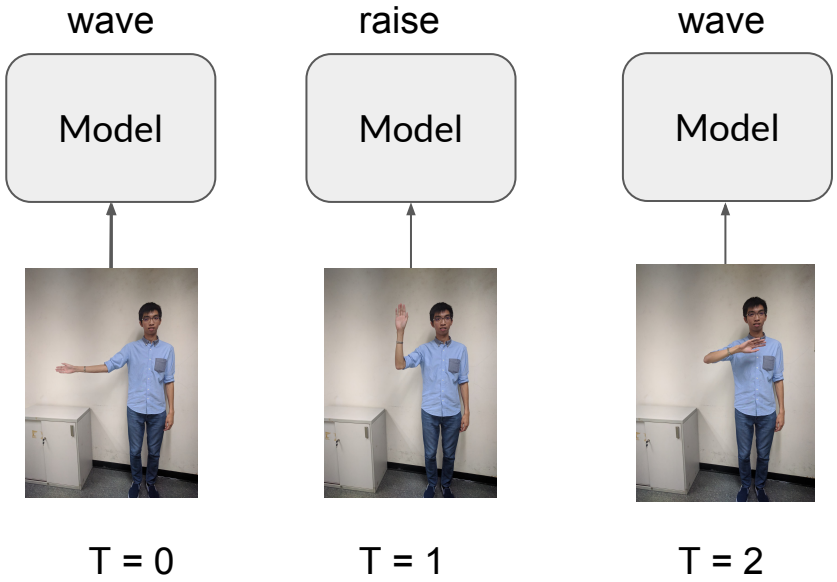
Key points: A video is a series of images!

The model should:

- Be able to handle images
- Has the concept of time

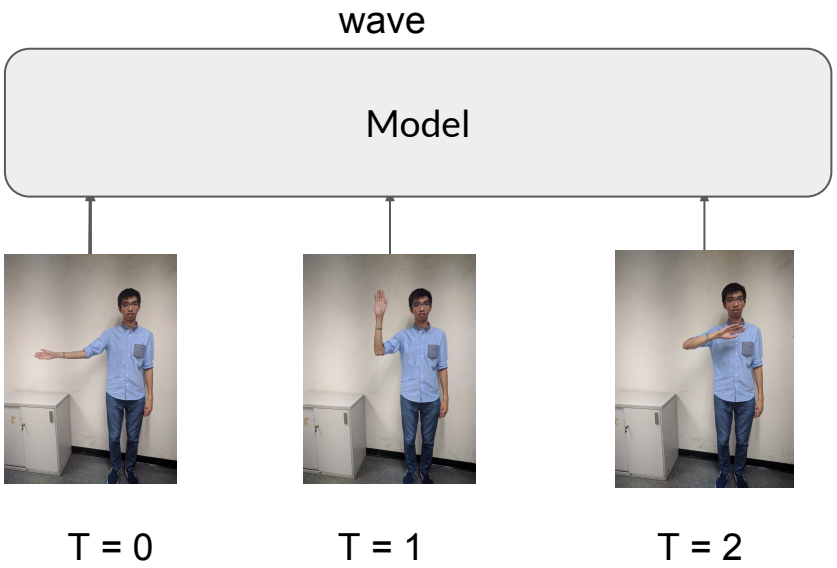
Action recognition with voting

- Deep Neural Network (DNN) framework on each frame



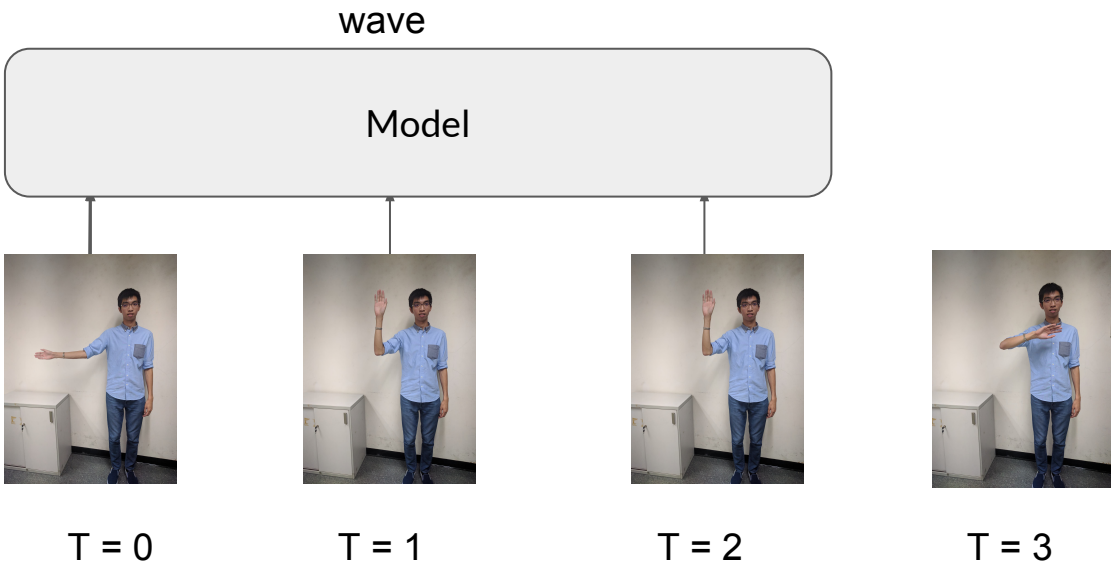
Single fully connected layer

- Deep Neural Network (DNN) framework on full sequence



Recurrent Neural Network (RNN)

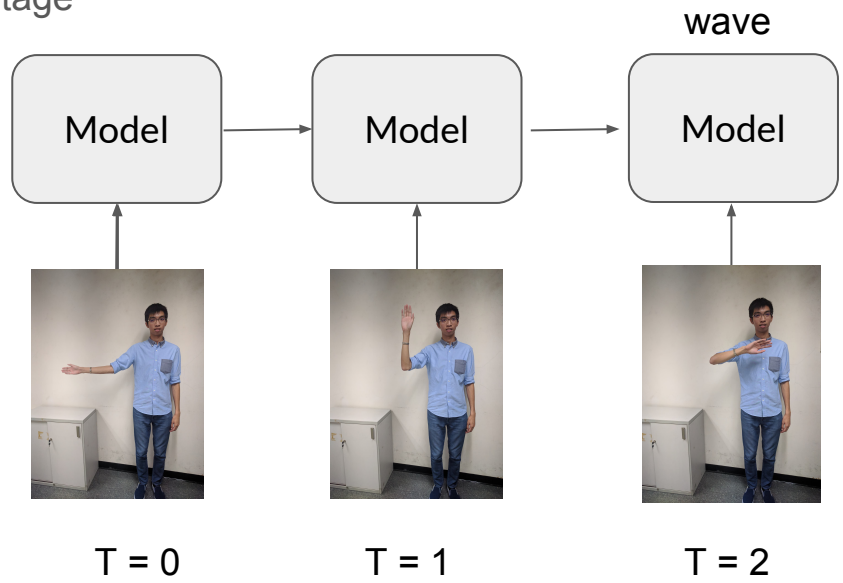
- Deep Neural Network (DNN) framework on full sequence



Problem: need a way to handle sequence of different length

Recurrent Neural Network (RNN)

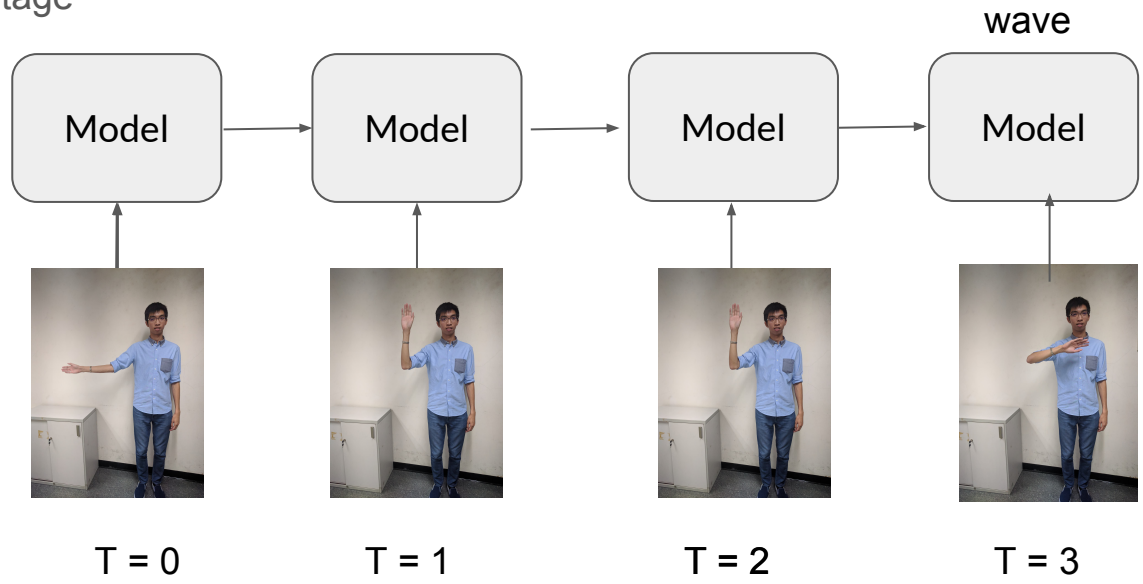
- Recurrent Neural Network remembers the past by passing previous information as input to the next stage



New input feature = [original input feature, output of the layer at previous time step]

Recurrent Neural Network (RNN)

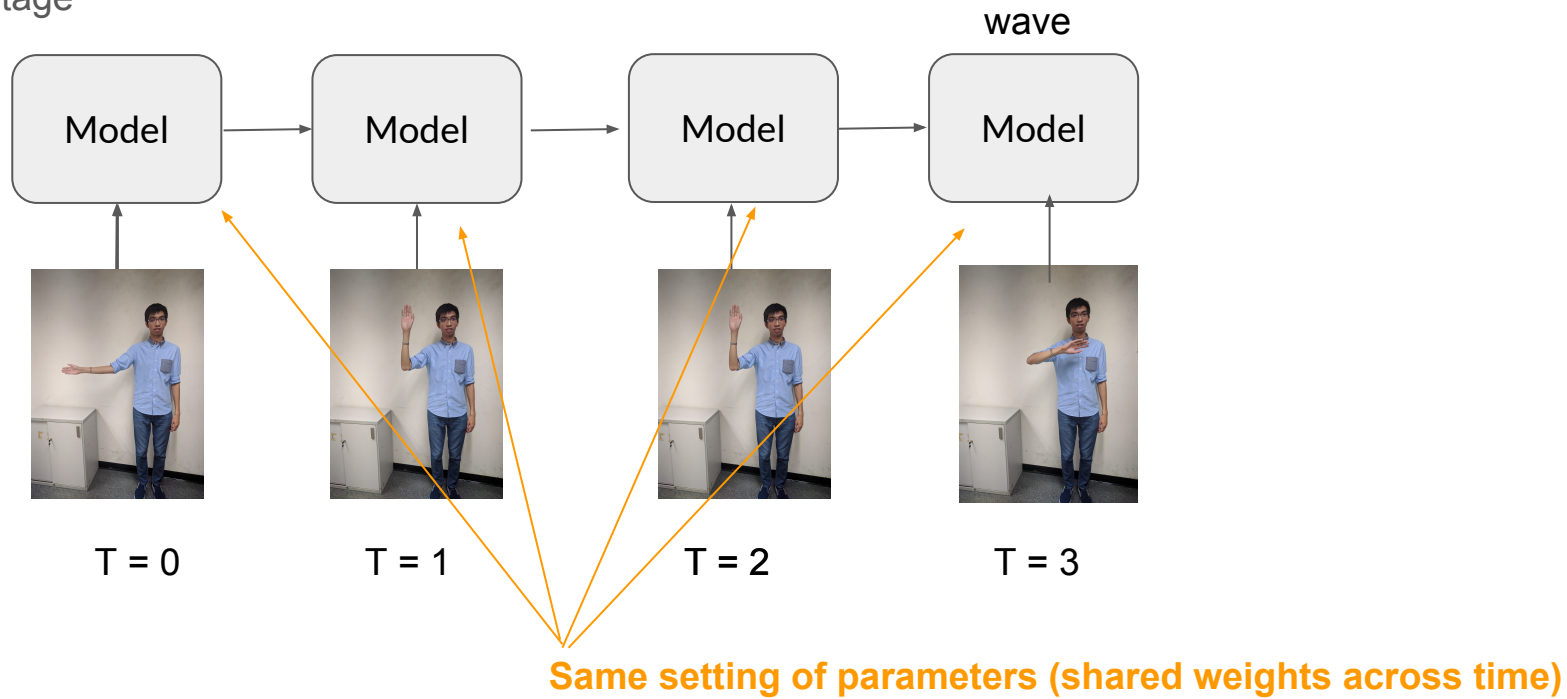
- Recurrent Neural Network remembers the past by passing previous information as input to the next stage



New input feature = [original input feature, output of the layer at previous time step]

Recurrent Neural Network (RNN)

- Recurrent Neural Network remembers the past by passing previous information as input to the next stage



Long Short-Term Memory (LSTM)

- Mostly used version of recurrent layer. Can choose to **remember**, **forget**, and **output** information

j is the index of the LSTM cell

$$o_t^j = \sigma\left(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + V_o \mathbf{c}_t\right)^j$$

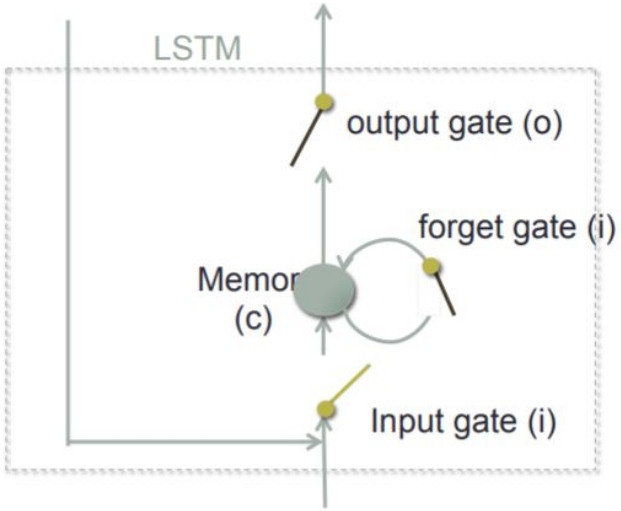
$$f_t^j = \sigma\left(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + V_f \mathbf{c}_{t-1}\right)^j$$

$$i_t^j = \sigma\left(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + V_i \mathbf{c}_{t-1}\right)^j .$$

$$\tilde{c}_t^j = \tanh\left(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1}\right)^j$$

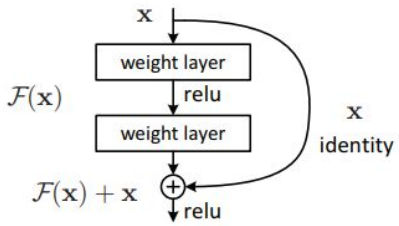
$$c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j$$

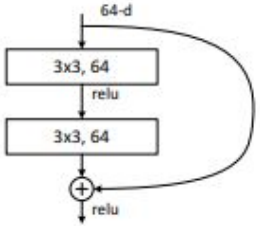
$$o_t^j = \sigma\left(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + V_o \mathbf{c}_t\right)^j$$

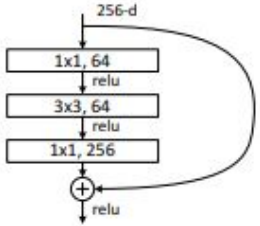


ResNet (CNN)

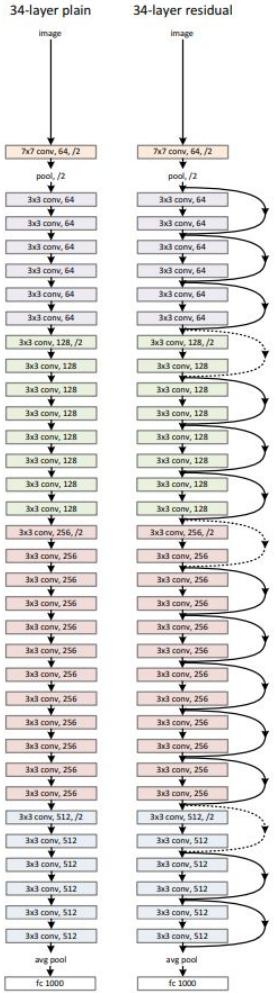
- Gradient vanishing makes it hard to train CNN with many layers
  - Use residual connection



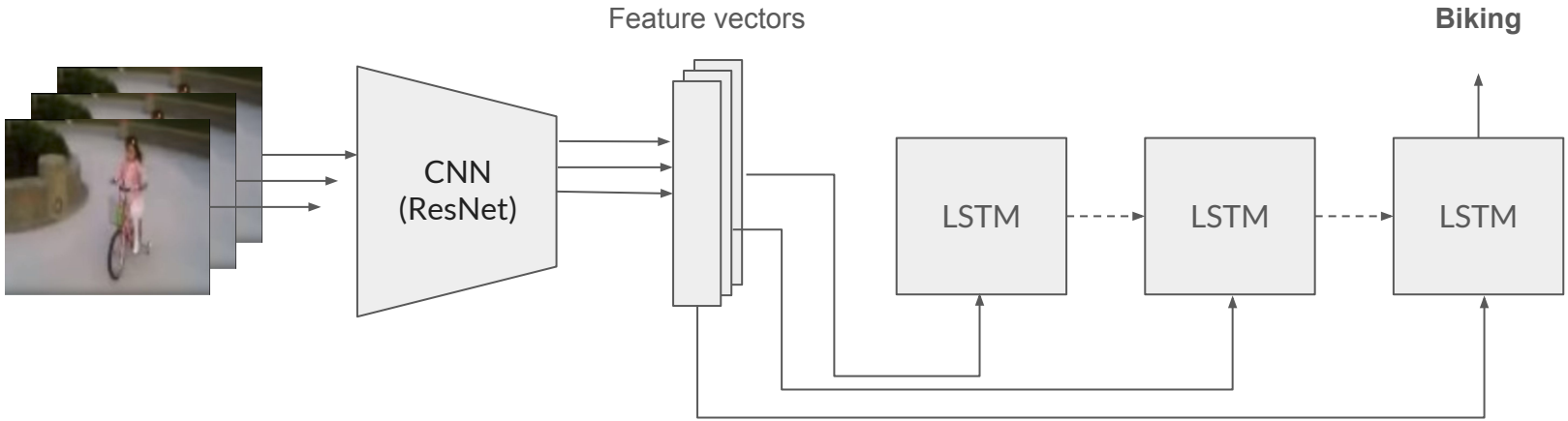




He, et al. 2016

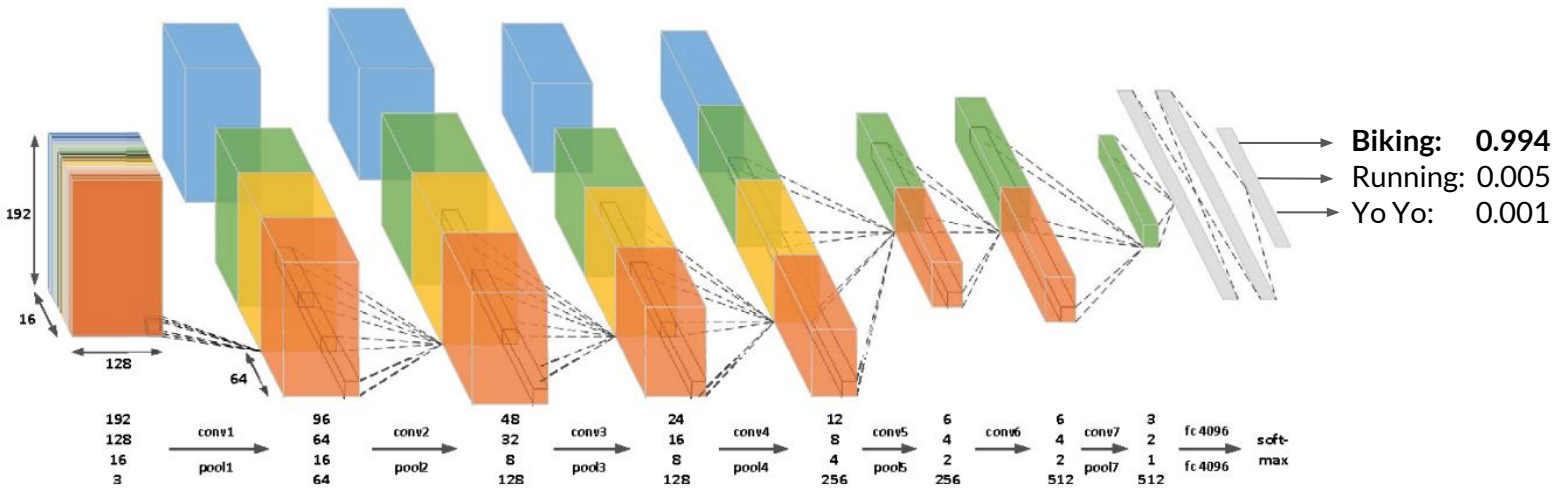


ResNet + LSTM



3D Convolutional Neural Networks

- Concatenate the images along time axis to get a 4-D data (3D from image [255x255x3] + 1D from time)
  - We can think of the whole video as a block of 4D input and do 3D convolution on it



Wolf, Thomas et al. 2016

Lab 4: Overview

In this lab, we will explore a few options on how to do action recognition on UCF 101 dataset, including:

- Create a baseline by doing image classification on only one sampled frame of the videos
  - Use ResNet as a base model for transfer learning. Train only the last layer on out data
  - Unfreeze some layers and train further
- Build a CNN + LSTM model for video input
  - Use ResNet as a feature extractor then feed the inputs to LSTM layers
- Discuss other possible model and performance of each one, including

  [CNN+LSTM trained from scratch, 3D convolutional neural networks, 3D CNN+dense layers]