CVPR 2018
Salt Lake City

Temporal Reasoning in Videos using Convolutional Gated Recurrent Units

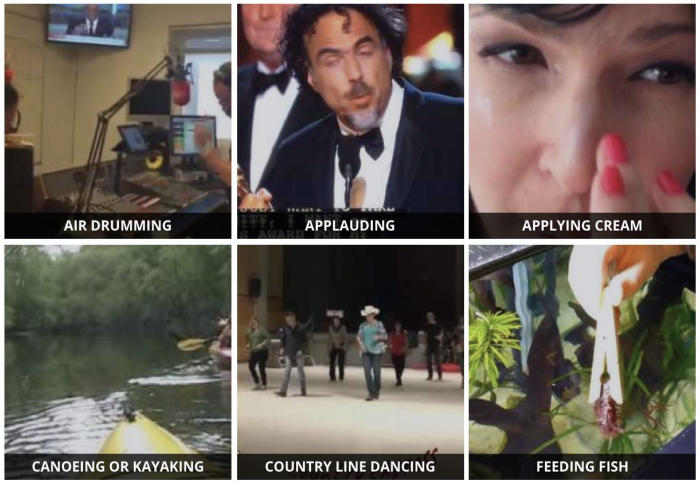
Debidatta Dwibedi*
Pierre Sermanet
Jonathan Tompson

Google

* Work done as a part of the Google AI Residency

Problem Setting

Action Recognition in Videos



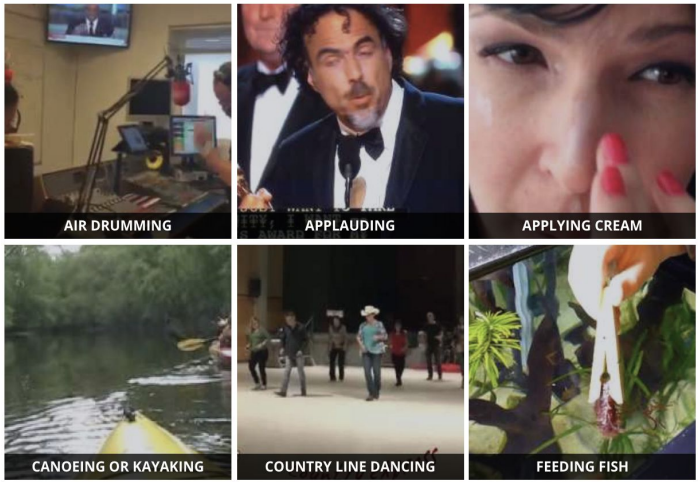
Kinetics[1]

UCF101[2]

[1] Kay et al., The Kinetics Human Action Video Dataset
[2] Soomro et al., UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild

Problem Setting

Action Recognition in Videos

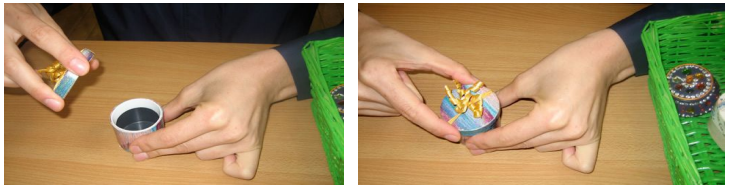


It is <u>possible</u> to predict correct action correctly from single frame

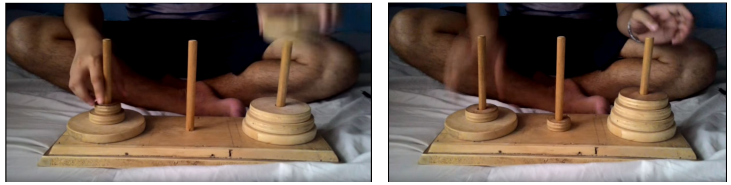
Problem Setting

Human-Object Interaction Recognition in Videos

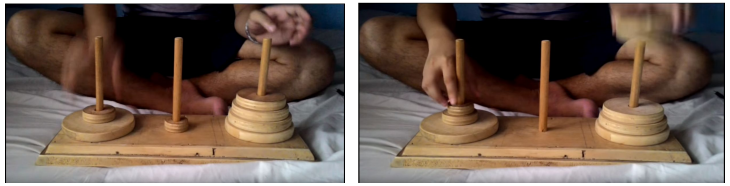
Opening



Closing



Transfer from left to middle



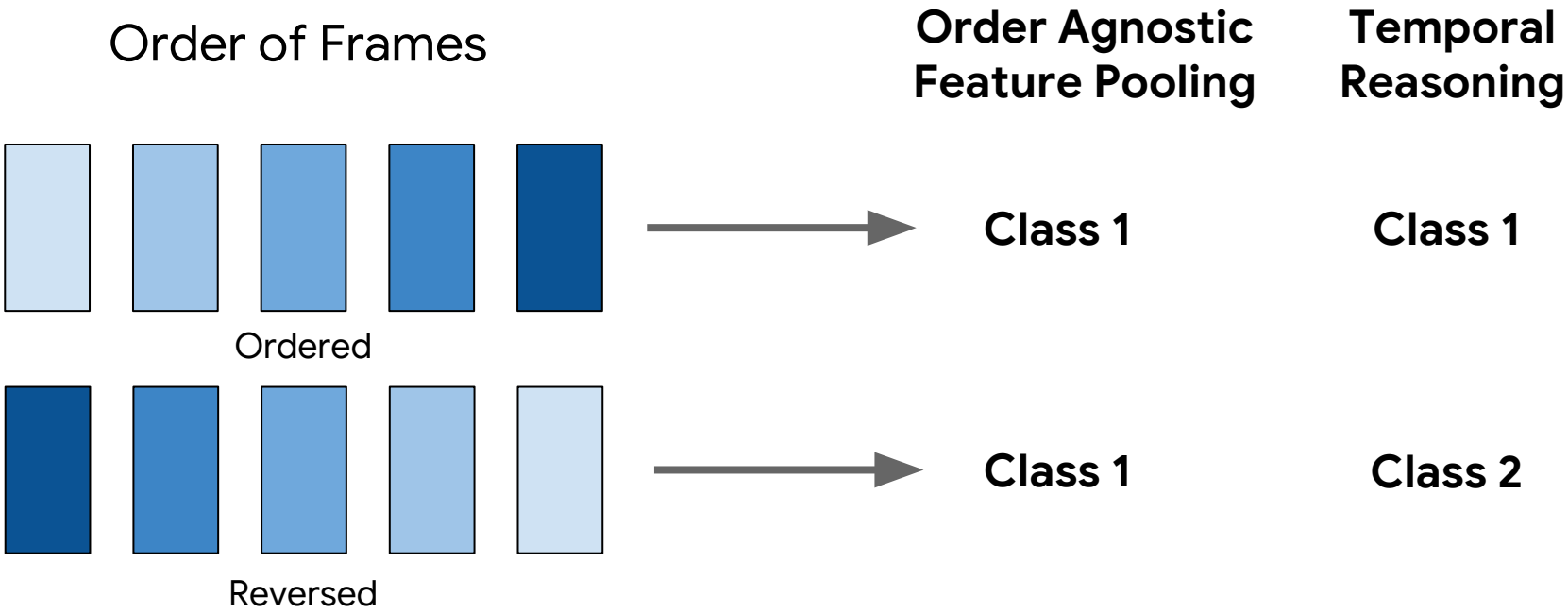
Transfer from middle to left

Impossible to predict correct actions when only one frame is input

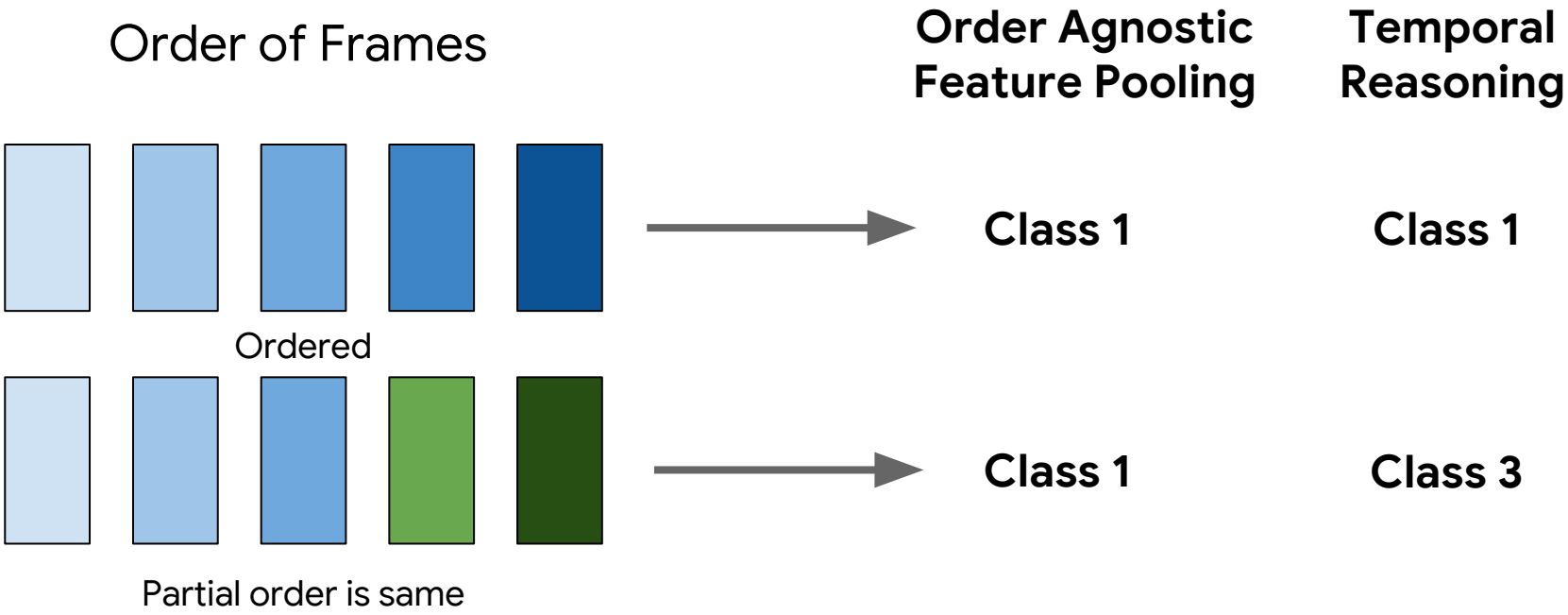
Question 1

Are all action recognition from video problems equivalent?

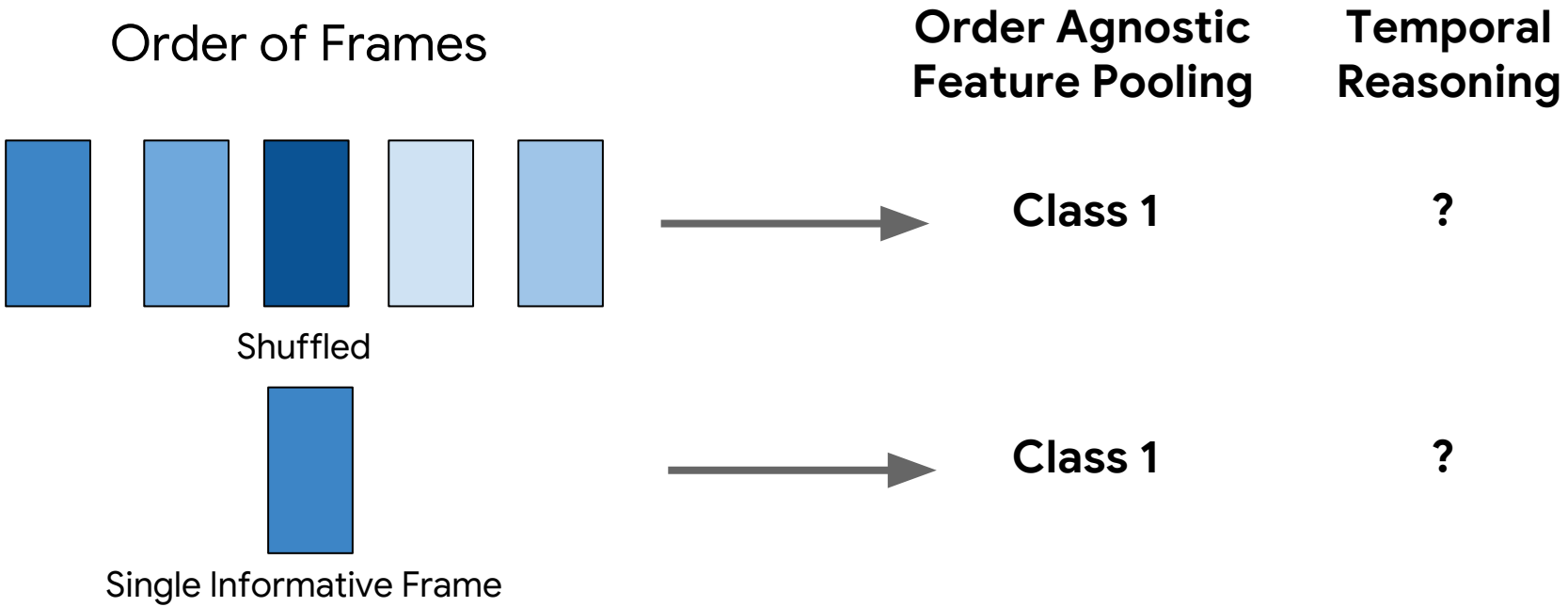
Categories of Action Recognition Problems



Categories of Action Recognition Problems



Categories of Action Recognition Problems

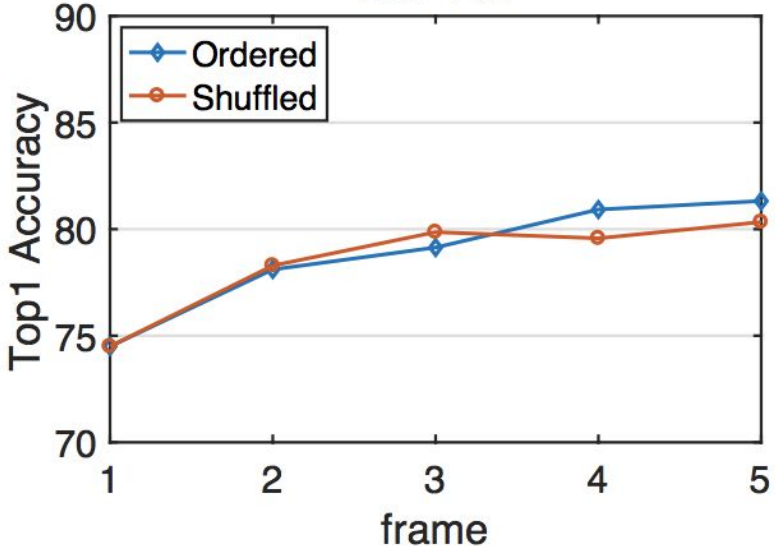


Categories of Action Recognition Problems

1. Difficult to categorize whole datasets into one category
2. Easier to test how much a dataset is biased towards one category

Categories of Action Recognition Problems

UCF101 [3]



Kinetics [4]

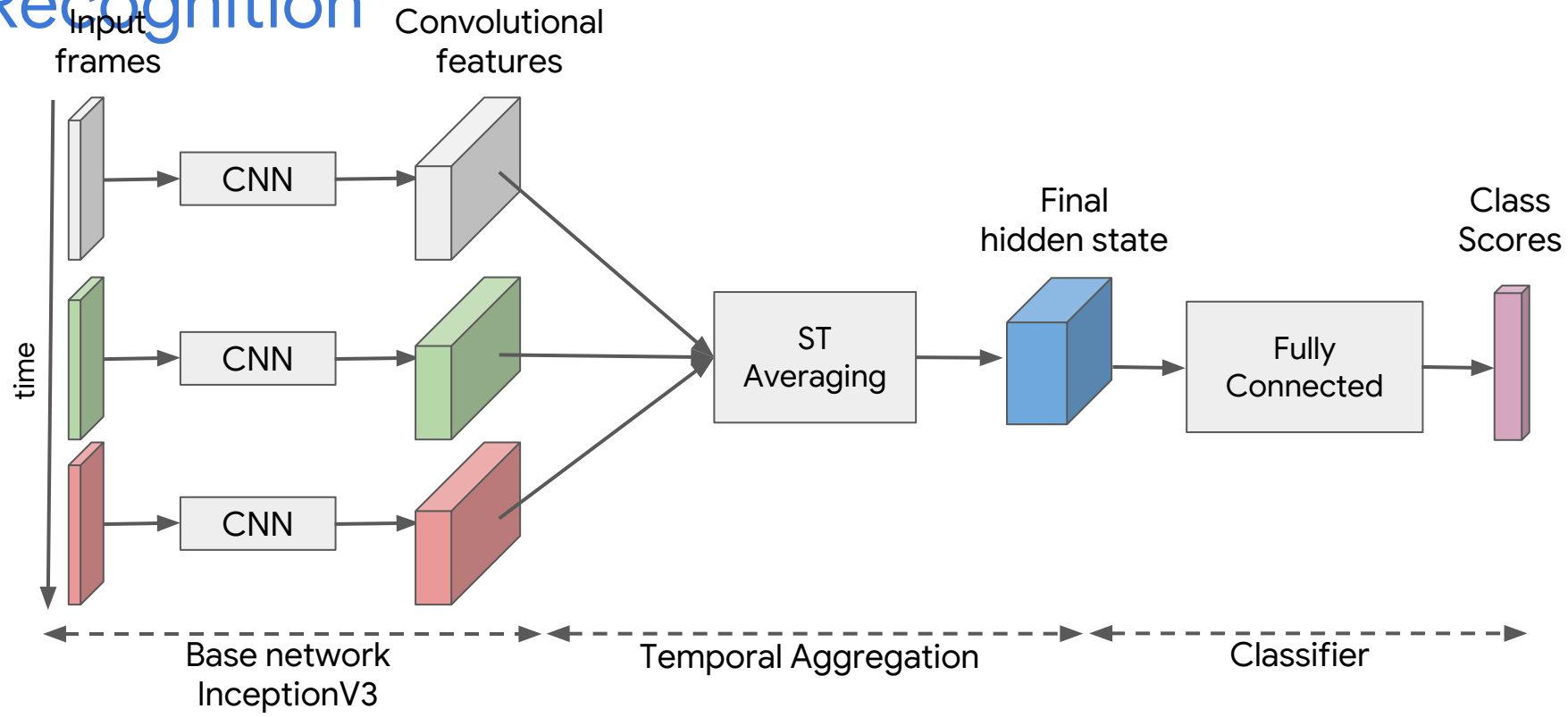
Model	Normal (%)	Shuffled (%)	Reversed (%)
I3D	71.66	45.37	71.54
I2D	67.00	67.52	67.23

[3] Zhuo et al., Temporal Relational Reasoning in Videos
[4] Xie et al., Rethinking Spatiotemporal Feature Learning For Video Understanding

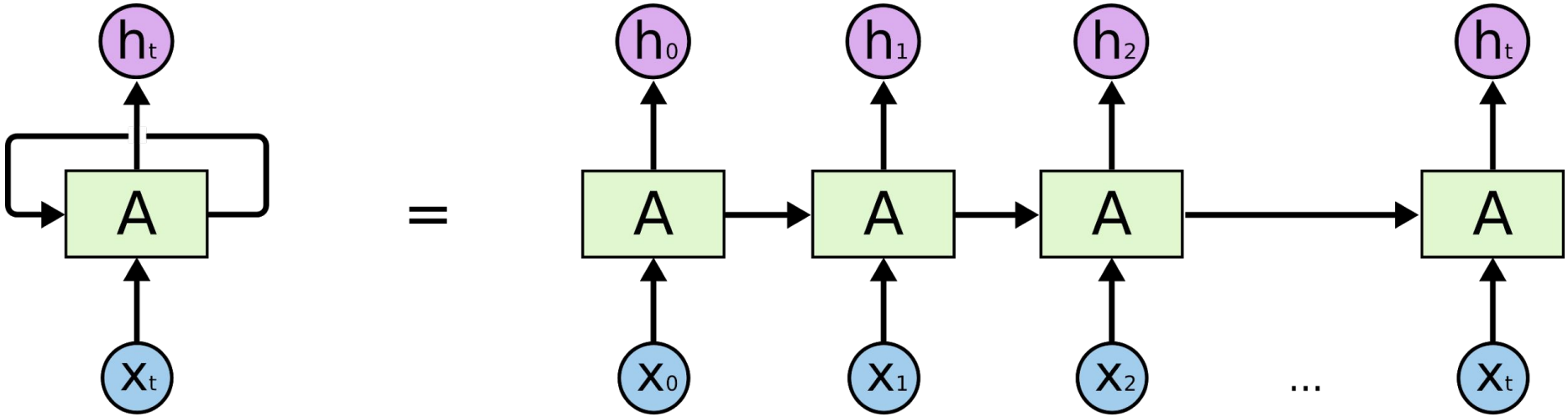
Question 2

Will there be one architecture for all action recognition tasks?

Spatio-temporal Averaging for Action Recognition

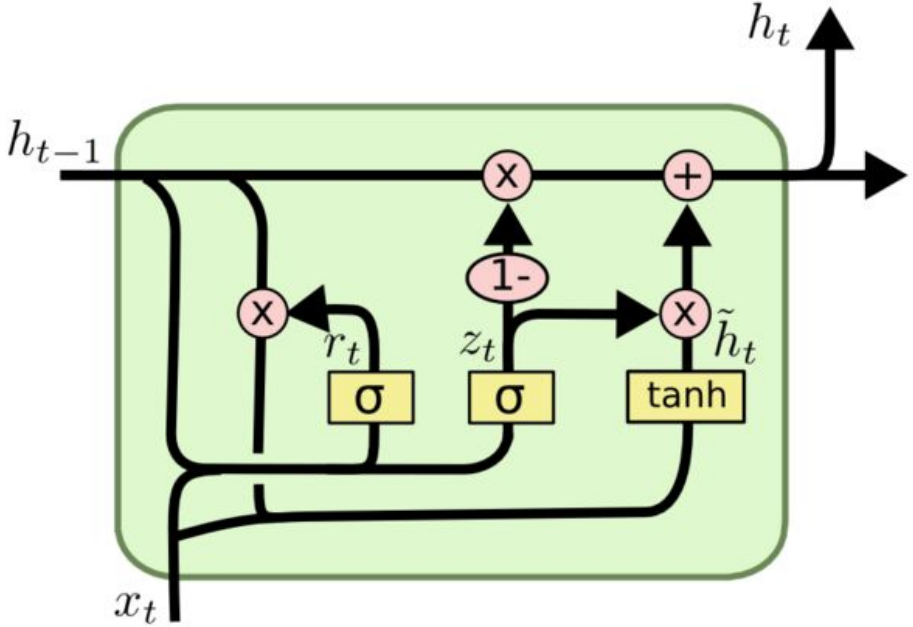


RNNs Recap



From http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Unit



From http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Equations

$$z_t = \sigma(W_{hz} h_{t-1} + W_{xz} x_t + b_z)$$
$$r_t = \sigma(W_{hr} h_{t-1} + W_{xr} x_t + b_r)$$
$$\hat{h_t} = \Phi(W_h (r_t \odot h_{t-1}) + W_x x_t + b)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z \odot \hat{h_t}$$

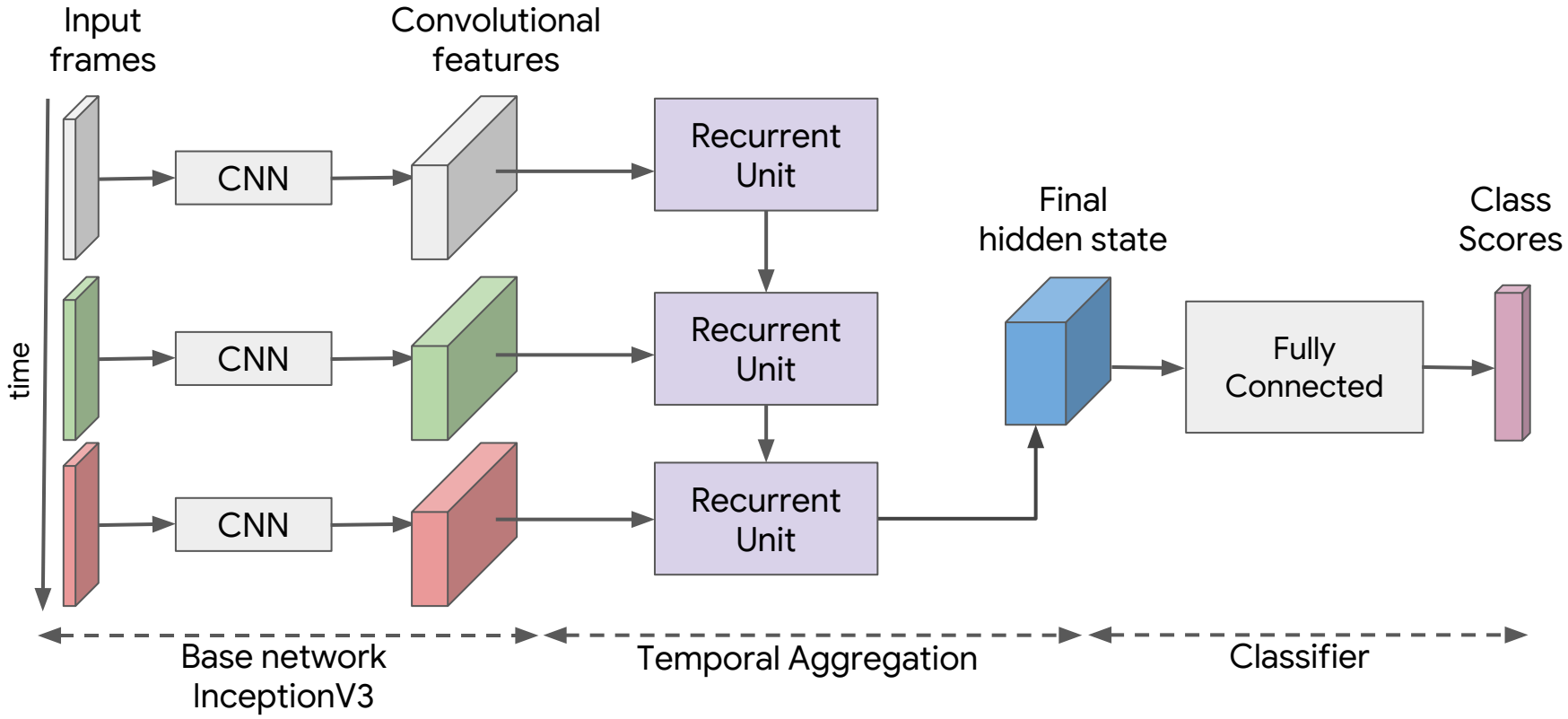
GRU

$$z_t = \sigma(W_{hz} * h_{t-1} + W_{xz} * x_t + b_z)$$
$$r_t = \sigma(W_{hr} * h_{t-1} + W_{xr} * x_t + b_r)$$
$$\hat{h_t} = \Phi(W_h * (r_t \odot h_{t-1}) + W_x * x_t + b)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z \odot \hat{h_t}$$

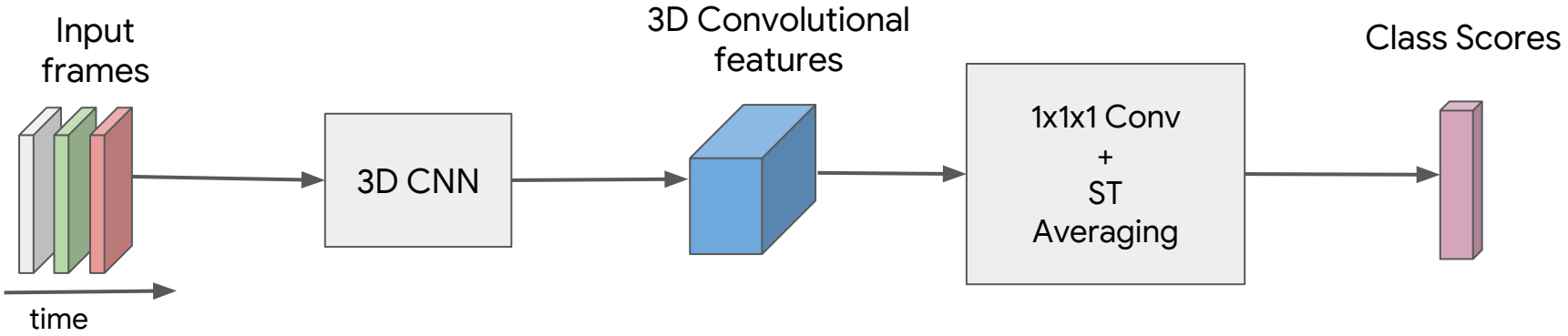
ConvGRU

Siam et al., Convolutional Gated Recurrent Networks for Video Segmentation

RNNs for Action Recognition (Ours)



I3D[5] (Current SoTA/Base network of SoTA models on many datasets)



[5] Carreira and Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset

Results on Kinetics

Method	Accuracy @ 1	Accuracy @ 5
Spatio-temporal Averaging	71.5	89.5
GRU	70.6	88.4
ConvGRU	70.0	88.1
I3D	**71.6**	**90.2**

Recurrent Units do not provide performance boosts when problem is not sequential

Results on Human-object Interactions

Method	Accuracy @ 1	Accuracy @ 5
Spatio-temporal Averaging	20.5	48.2
GRU	35.4	63.3
I3D (Kinetics pre-trained)	39.9	67.8
ConvGRU	43.7	71.4
ConvGRU (Large)	**45.9**	**74.5**

Recurrent Units are effective for temporal reasoning tasks

Question 3

What is encoded in the hidden state of the recurrent units?

Action-Conditioned Nearest Neighbors

We want to get insight into what is being encoded in the hidden states.

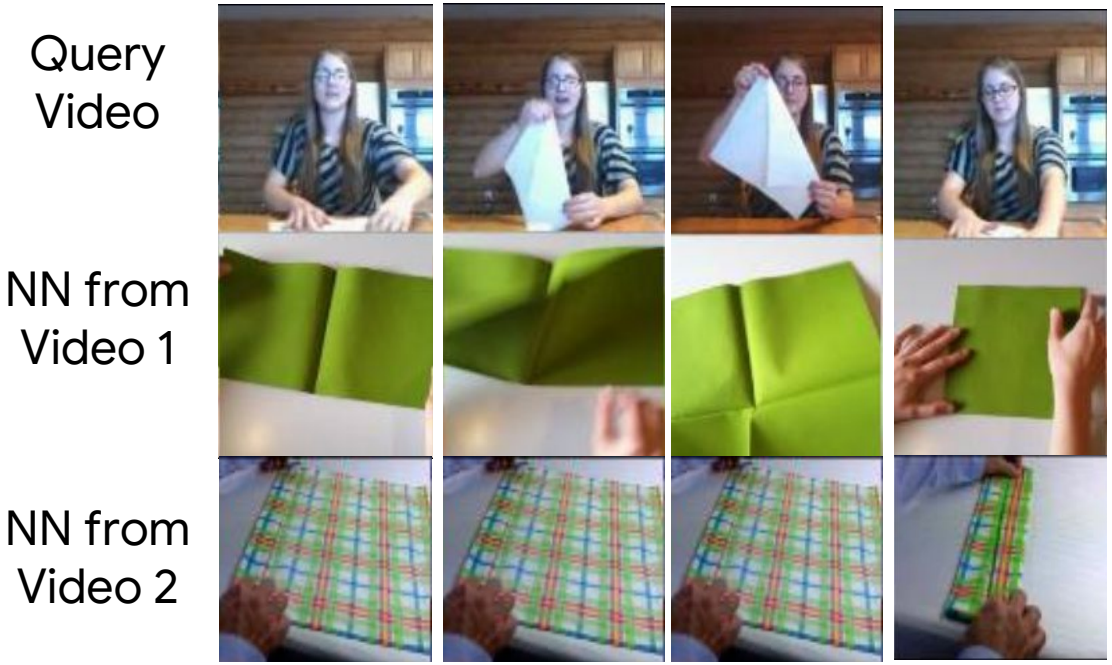
Experiment:

1) Choose 2 videos from a given class
2) Embed both videos using ConvGRU model
3) For each frame in video 1 look up nearest neighbours in video 2

Action-Conditioned Nearest Neighbors

Flipping Pancake

Action-Conditioned Nearest Neighbors



Folding Napkins

Action-Conditioned Nearest Neighbors

Alignment more prominent in human-object interactions

Positions of objects, states of objects (open/closed, filled/empty), abstract concepts like count are encoded

Tricks for using RNNs effectively

1) Randomly sample frames in video
2) Layer normalization
3) Dropout at each timestep
4) Reduce number of features in the RNN
5) Reduce number of features in input (using bottlenecks) to RNN

Key Takeaways

1) All action recognition tasks are not equivalent
2) Depending on the task, different architectures are useful
3) RNNs more useful if the task of action recognition is sequential in nature
4) Hidden states capture interesting transitions for temporal reasoning tasks

The End