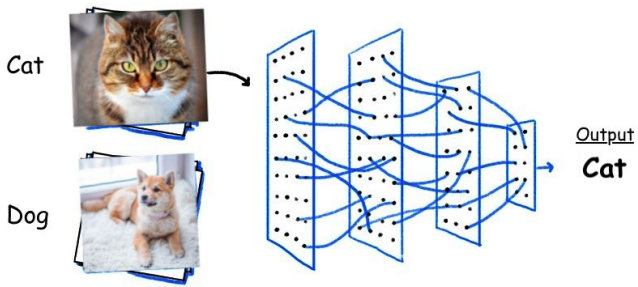
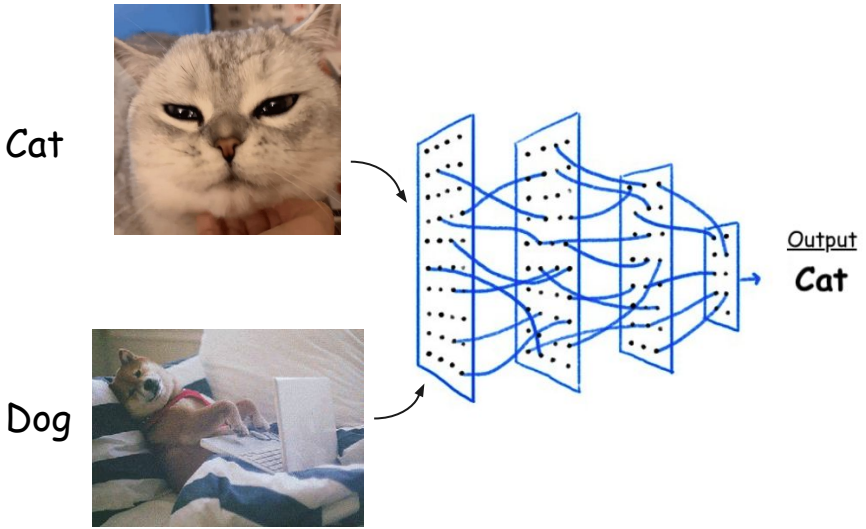
Video Classification

Deyu Kong, Aihan Liu

Instructor: Prof. Amir Jafari
DATS6203 Machine Learning 2

Image Classification vs Video Classification





Video Classification Methods

2014	Two-Stream Networks	https://papers.nips.cc/paper/2014/file/00ec53c4682d36f5c4359f4ae7bd7ba1-Paper.pdf *(NeurIPS Neural Information Processing Systems)*.	Two-Stream convolutional neural network is proposed to extract spatio-temporal information
2015	C3D	https://arxiv.org/abs/1412.0767 *(IEEE international conference on computer vision)*.	3D Convolutional Network

3D CNN: (Batch, Channel, Frame, Height, Width)

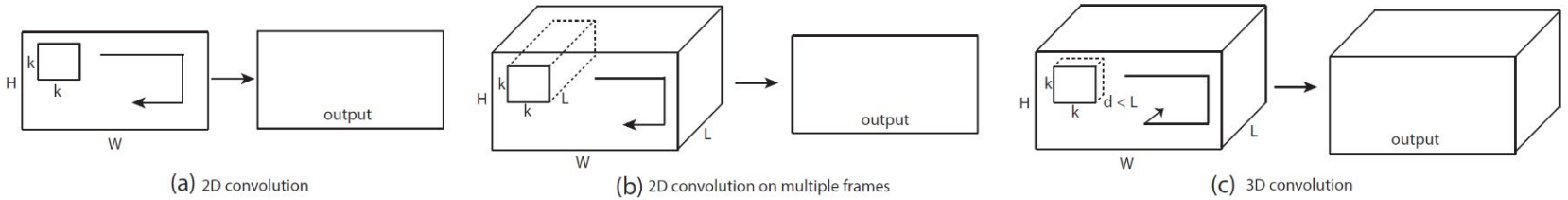


Figure 1. **2D and 3D convolution operations**. a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal.

3D CNN: (Batch, Channel, Frame, Height, Width)

CLASS `torch.nn.Conv3d(`*`in_channels`*`,` *`out_channels`*`,` *`kernel_size`*`,` *`stride=1`*`,` *`padding=0`*`,`
`dilation=1``,` *`groups=1`*`,` *`bias=True`*`,` *`padding_mode='zeros'`*`,` *`device=None`*`,` *`dtype=None`*`)` [SOURCE]

Applies a 3D convolution over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C_{in}, D, H, W) and output $(N, C_{out}, D_{out}, H_{out}, W_{out})$ can be precisely described as:

$$out(N_i, C_{out_j}) = bias(C_{out_j}) + \sum_{k=0}^{C_{in}-1} weight(C_{out_j}, k) \star input(N_i, k)$$

Dataset - UCF101

UCF101 contains 101 categories and 13,320 videos recorded in unconstrained environments and uploaded to YouTube featuring camera motion, various lighting conditions, partial occlusion, low-quality frames, and more.

Actions	101
Clips	13320
Groups per Action	25
Clips per Group	4-7
Mean Clip Length	7.21sec
Total Duration1	1600mins
Min Clip Length	1.06sec
Max Clip Length	71.04sec
Frame Rate	25 fps
Resolution	320 x 240
Audio	Yes

Playing Instruments

Playing Guitar

Playing Piano

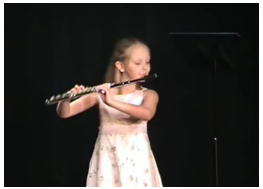
Playing Tabla

Playing Violin

Playing Cello

Playing Daf

Playing Dhol



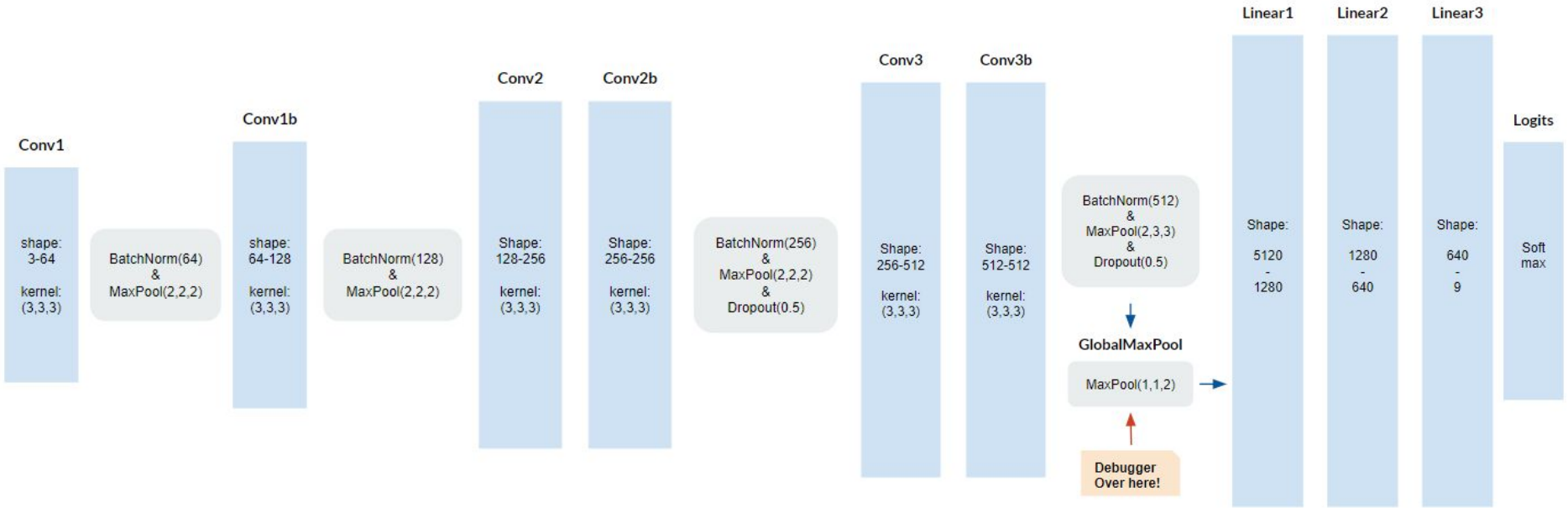
Playing Flute

Playing Sitar

Baseline:

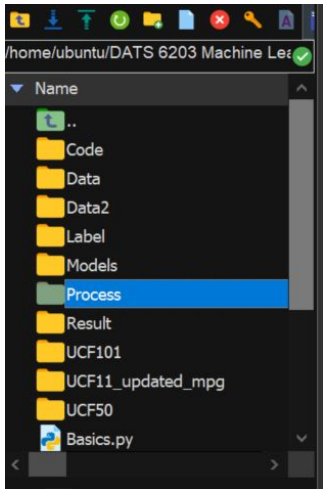
37.42%

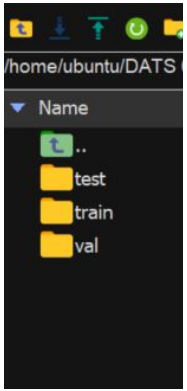
Model-VC3D

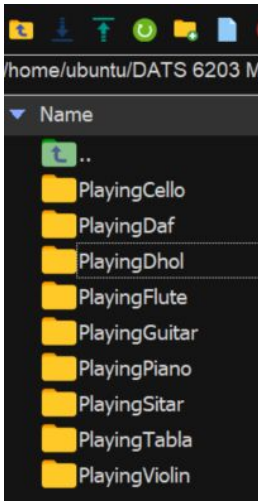

Conv1
shape:
3-64
kernel:
(3,3,3)
BatchNorm(64)
&
MaxPool(2,2,2)
Conv1b
shape:
64-128
kernel:
(3,3,3)
BatchNorm(128)
&
MaxPool(2,2,2)
Conv2
Shape:
128-256
kernel:
(3,3,3)
Conv2b
Shape:
256-256
kernel:
(3,3,3)
BatchNorm(256)
&
MaxPool(2,2,2)
&
Dropout(0.5)
Conv3
Shape:
256-512
kernel:
(3,3,3)
Conv3b
Shape:
512-512
kernel:
(3,3,3)
BatchNorm(512)
&
MaxPool(2,3,3)
&
Dropout(0.5)
GlobalMaxPool
MaxPool(1,1,2)
Debugger
Over here!
Linear1
Shape:
5120
-
1280
Linear2
Shape:
1280
-
640
Linear3
Shape:
640
-
9
Logits
Soft
max

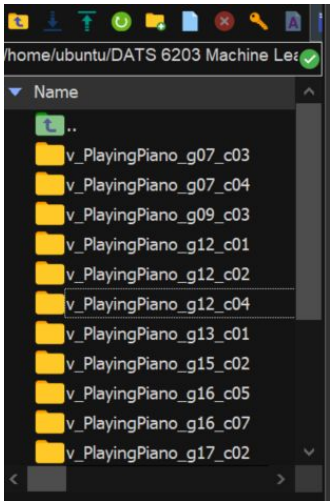
Dataloader

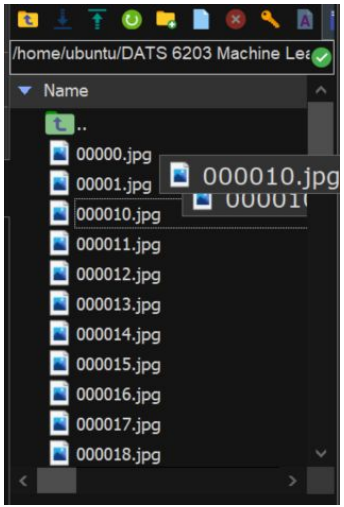
Process video as images, then resize to 171x128, then crop to 120x120











Training

BATCH_SIZE=20

N_EPOCH=50

LR=1e-3, ReduceLROnPlateau, monitor on test loss

Optimizer: Adam()

Criterion: CrossEntropyLoss() contains softmax inside

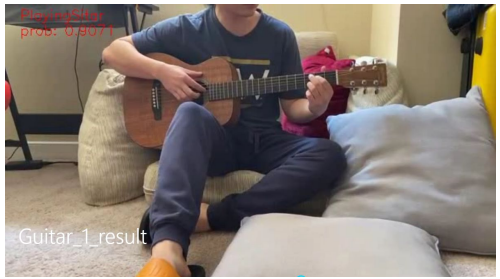
Metrics: accuracy - hamming #save model on this

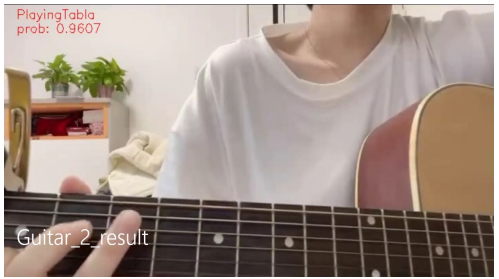
Perhaps include f1 score

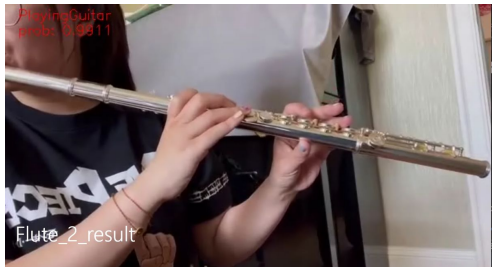
```
Epoch    25: reducing learning rate of group 0 to 1.5625e-05.
Epoch 25: 100%|██████████| 44/44 [00:44<00:00,  1.01s/it, Train Loss: 0.46177]
Epoch 25:  Train acc 0.85455 Train hlm -0.14545 Train sum 0.70909 Train avg 0.23636 -
Epoch 25: 100%|██████████| 10/10 [00:04<00:00,  2.42it/s, Test Loss: 0.44782]
Epoch 25:  Test acc 0.89119 Test hlm -0.10881 Test sum 0.78238 Test avg 0.26079 -
The model has been saved!
Trigger Times: 0
Epoch 26: 100%|██████████| 44/44 [00:44<00:00,  1.02s/it, Train Loss: 0.43973]
Epoch 26:  Train acc 0.86705 Train hlm -0.13295 Train sum 0.73409 Train avg 0.24470 -
Epoch 26: 100%|██████████| 10/10 [00:04<00:00,  2.49it/s, Test Loss: 0.43799]
Epoch 26:  Test acc 0.90155 Test hlm -0.09845 Test sum 0.80311 Test avg 0.26770 -
The model has been saved!
Trigger Times: 0
Epoch 27: 100%|██████████| 44/44 [00:44<00:00,  1.01s/it, Train Loss: 0.38911]
Epoch 27:  Train acc 0.87159 Train hlm -0.12841 Train sum 0.74318 Train avg 0.24773 -
Epoch 27: 100%|██████████| 10/10 [00:04<00:00,  2.43it/s, Test Loss: 0.47012]
Epoch 27:  Test acc 0.88601 Test hlm -0.11399 Test sum 0.77202 Test avg 0.25734 -
Trigger Times: 1
Epoch 28: 100%|██████████| 44/44 [00:44<00:00,  1.01s/it, Train Loss: 0.43277]
Epoch 28:  Train acc 0.84318 Train hlm -0.15682 Train sum 0.68636 Train avg 0.22879 -
Epoch 28: 100%|██████████| 10/10 [00:04<00:00,  2.45it/s, Test Loss: 0.37859]
Epoch 28:  Test acc 0.91192 Test hlm -0.08808 Test sum 0.82383 Test avg 0.27461 -
The model has been saved!
```

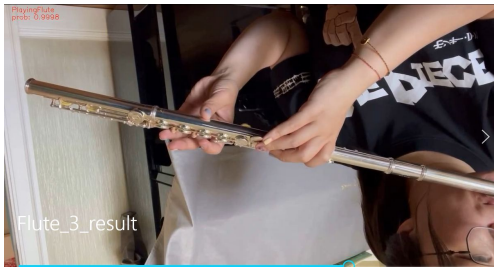
Test Result & Demo

Acc - Hlm: `Test acc 1.00000 Test hlm -0.00000 Test sum 1.00000`









Limitations & Potential Improvements

Misclassify Tabla, Guitar & Sitar: Not Robust enough

Apply smarter augmentation? E.g. Latent Space

RNN + Conv2D

Use attention mechanism (add tap delay line)

Apply Two-Stream Network: Temporal & Spatial (Mimic Human Vision Process)

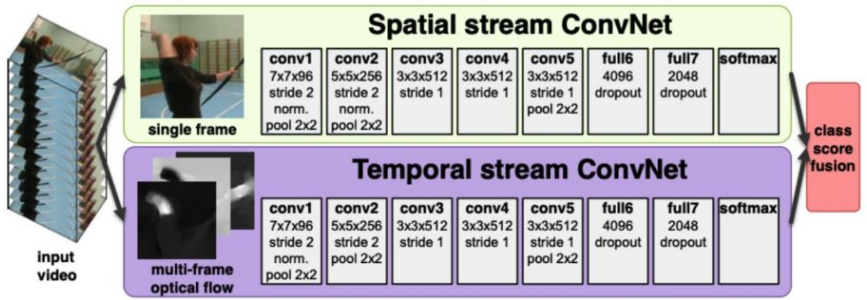


Figure 1: **Two-stream architecture for video classification.**

Acknowledgement

Mocha, who contributes his face for the cat video

Ian & Xu, who provide the material for guitar solo

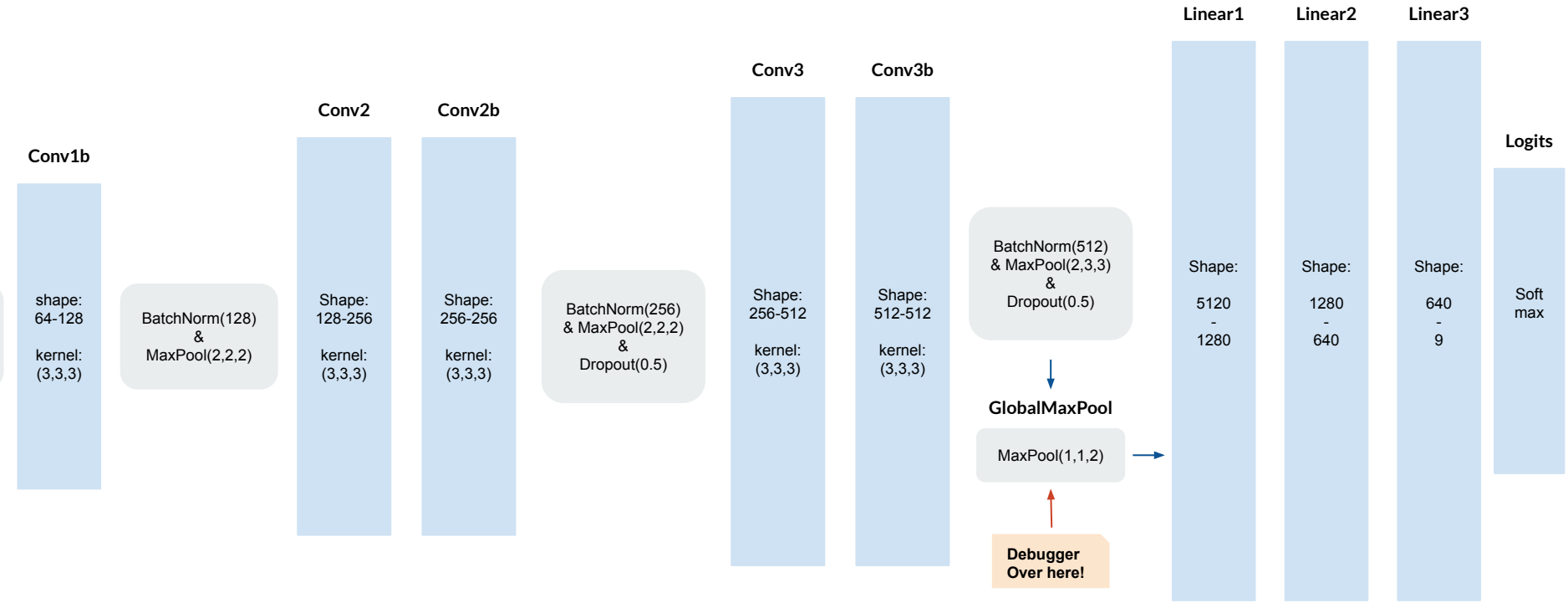
Tim, who provides the material for piano playing

Maggie Ye, who provides the material for Flute solo

References

- Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, *27*.
- Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 4489-4497).

Q & A


Conv1b
shape:
64-128
kernel:
(3,3,3)
BatchNorm(128)
&
MaxPool(2,2,2)
Conv2
Shape:
128-256
kernel:
(3,3,3)
Conv2b
Shape:
256-256
kernel:
(3,3,3)
BatchNorm(256)
& MaxPool(2,2,2)
&
Dropout(0.5)
Conv3
Shape:
256-512
kernel:
(3,3,3)
Conv3b
Shape:
512-512
kernel:
(3,3,3)
BatchNorm(512)
& MaxPool(2,3,3)
&
Dropout(0.5)
GlobalMaxPool
MaxPool(1,1,2)
Debugger
Over here!
Linear1
Shape:
5120
-
1280
Linear2
Shape:
1280
-
640
Linear3
Shape:
640
-
9
Logits
Soft
max