CREST hands-on meeting: MED baseline tutorial

CREST Deep 9th Meeting

Shinoda Lab.
Mengxi Lin
Ryosuke Yamamoto
Aug. 18, 2017

Today's goal

- Learn how our baseline works

  - We show you with slides

- Run the actual baseline system on TSUBAME 3.0

  - We will ask you to do several exercises

Agenda

1. MED introduction

   – Repeated: same as in 4th meeting

2. How to use GitHub

   – Clone the repository, throw an issue

3. How to extract features with Caffe

   – Use the baseline system, use a different CNN model, evaluate

4. How to train and test LSTM with Torch

   – Construct an LSTM net, extend the net, train the net, evaluate

Agenda

1. MED introduction

  – Repeated: same as in 4th meeting

2. How to use GitHub

3. How to extract features with Caffe

4. How to train and test LSTM with Torch

MED introduction

Agenda

1. MED introduction

2. How to use GitHub

   – Git / GitHub introduction

   – Clone the repository into a TSUBAME workspace

   – Throw an issue

3. How to extract features with Caffe

4. How to train and test LSTM with Torch

GitHub: Git / GitHub introduction

- Git is a version control system for codes

  - Briefly, it will detect differences between versions and recognize them as changes

- GitHub is a web service for developing a software with Git

  - We use it as just a code host

  - We update the code rarely

  - We maintain our code on our local Git, not on GitHub
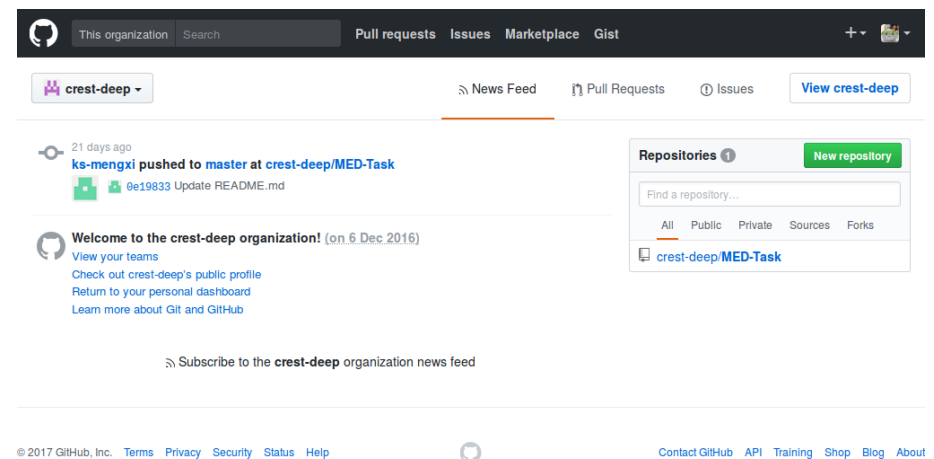
GitHub: Clone the repository

- Our repository is on https://github.com/crest-deep/MED-Task

- You can clone the repository by the following command:

```
> git clone https://github.com/crest-deep/MED-Task.git
```

- Unfortunately, TSUBAME 3.0 doesn't have git environment

  - You'll need the following command to get a copy of the repository:

```
> wget
https://github.com/crest-deep/MED-Task/archive/hands-on.zip
> unzip hands-on.zip
```

GitHub: *Exercise*

- Clone our repository into your TSUBAME workspace

- Our repository is on https://github.com/crest-deep/MED-Task

- Log-in to TSUBAME 3.0:

```
> ssh -Y -i [rsa_key] [id]@login.t3.gsic.titech.ac.jp
```

- You can create your workspace by the following:

```
> cd /gs/hs0/tga-crest-deep/
> mkdir $USER
> cd $USER
```

- Use the following command to get a copy of the repository:

```
> wget https://github.com/crest-deep/MED-Task/archive/master.zip
> unzip master.zip
```

GitHub: Throw an issue

- You can notify us of issues such as bugs, typos and unclear points of the repository by GitHub Issues feature

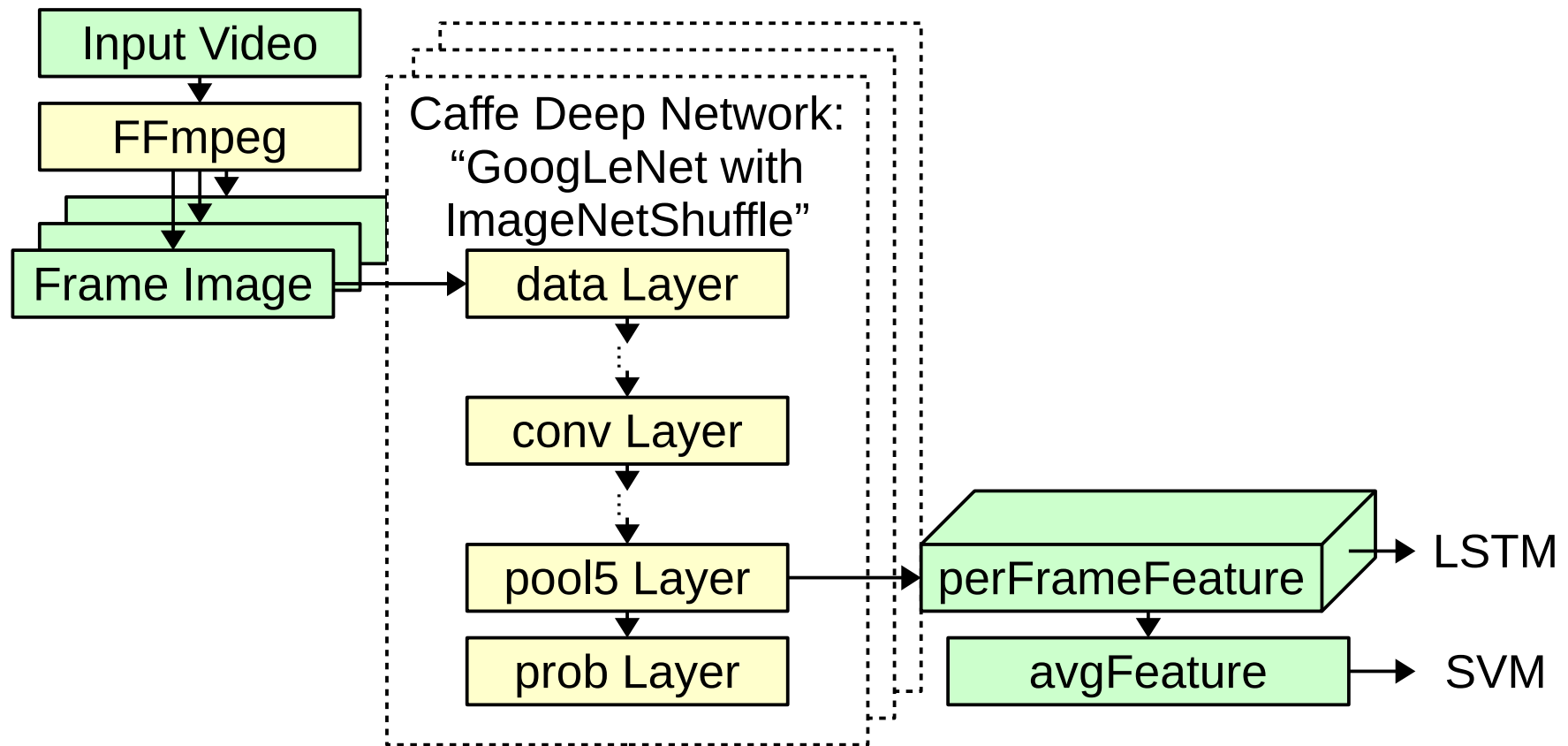 - You'll need a GitHub account in order to throw an issue
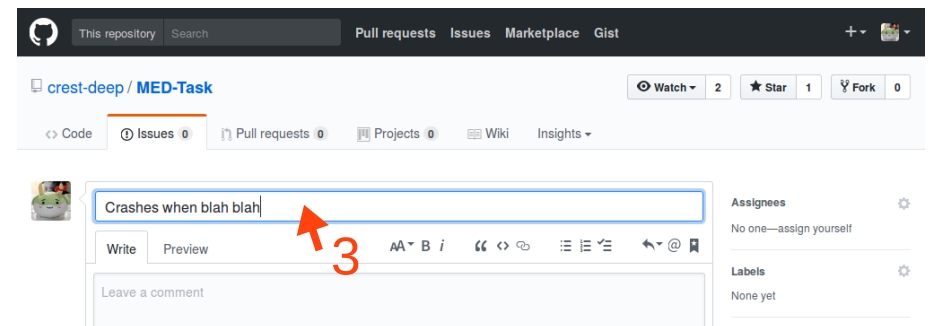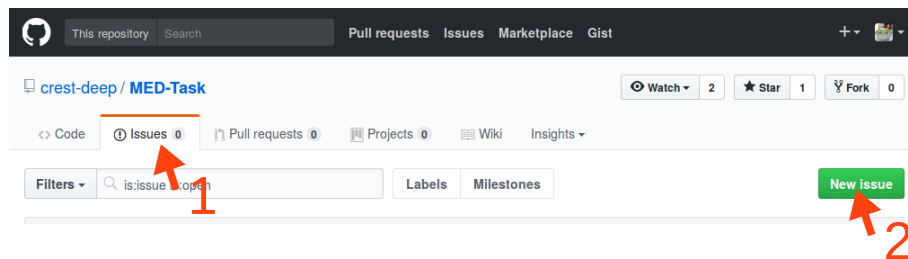
- You can also ask us any questions via Slack CREST team

Agenda

1. MED introduction

2. How to use GitHub

3. How to extract features with Caffe

 – Extract features by using the baseline system

 – Extract features with a different CNN model / layer

 – Evaluate performance with SVM

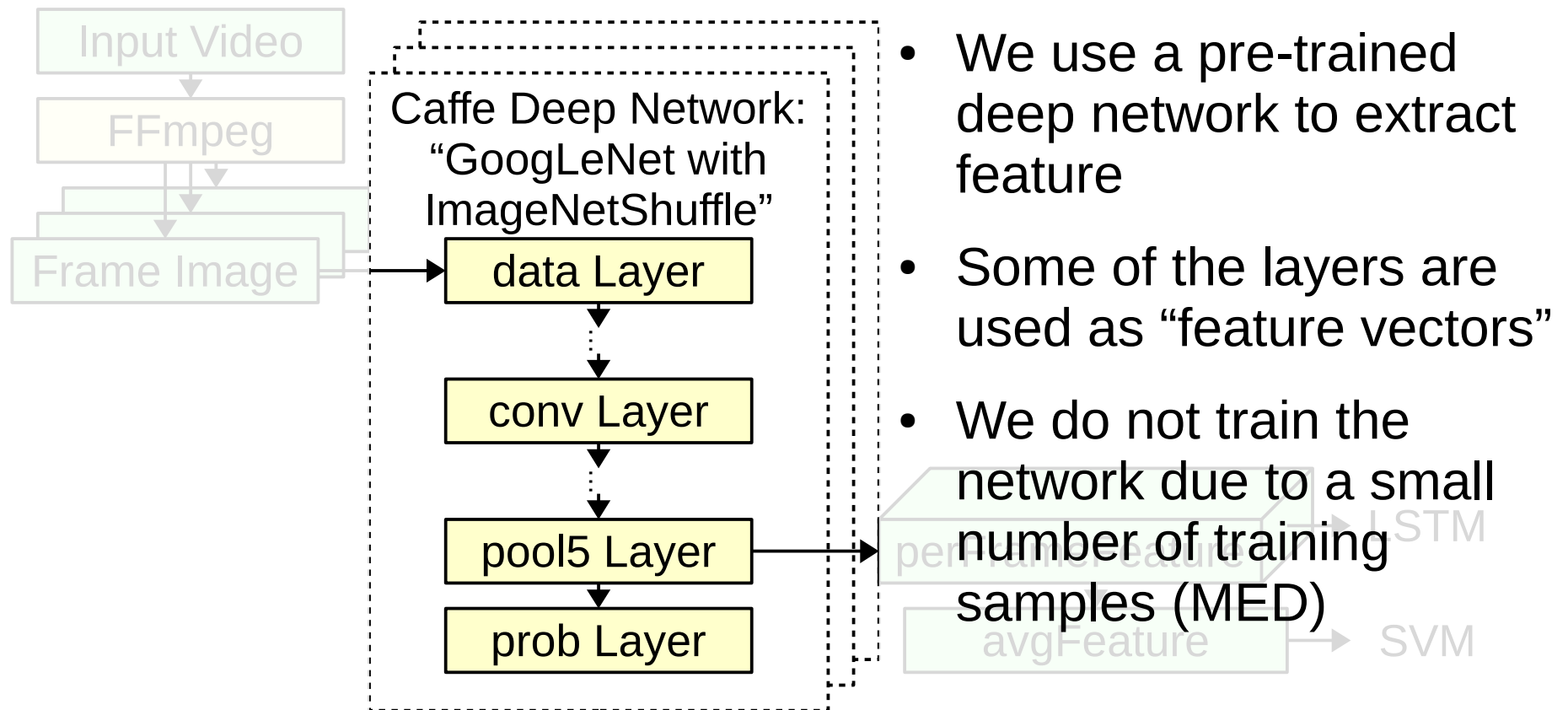4. How to train and test LSTM with Torch

Extract feature: Overview
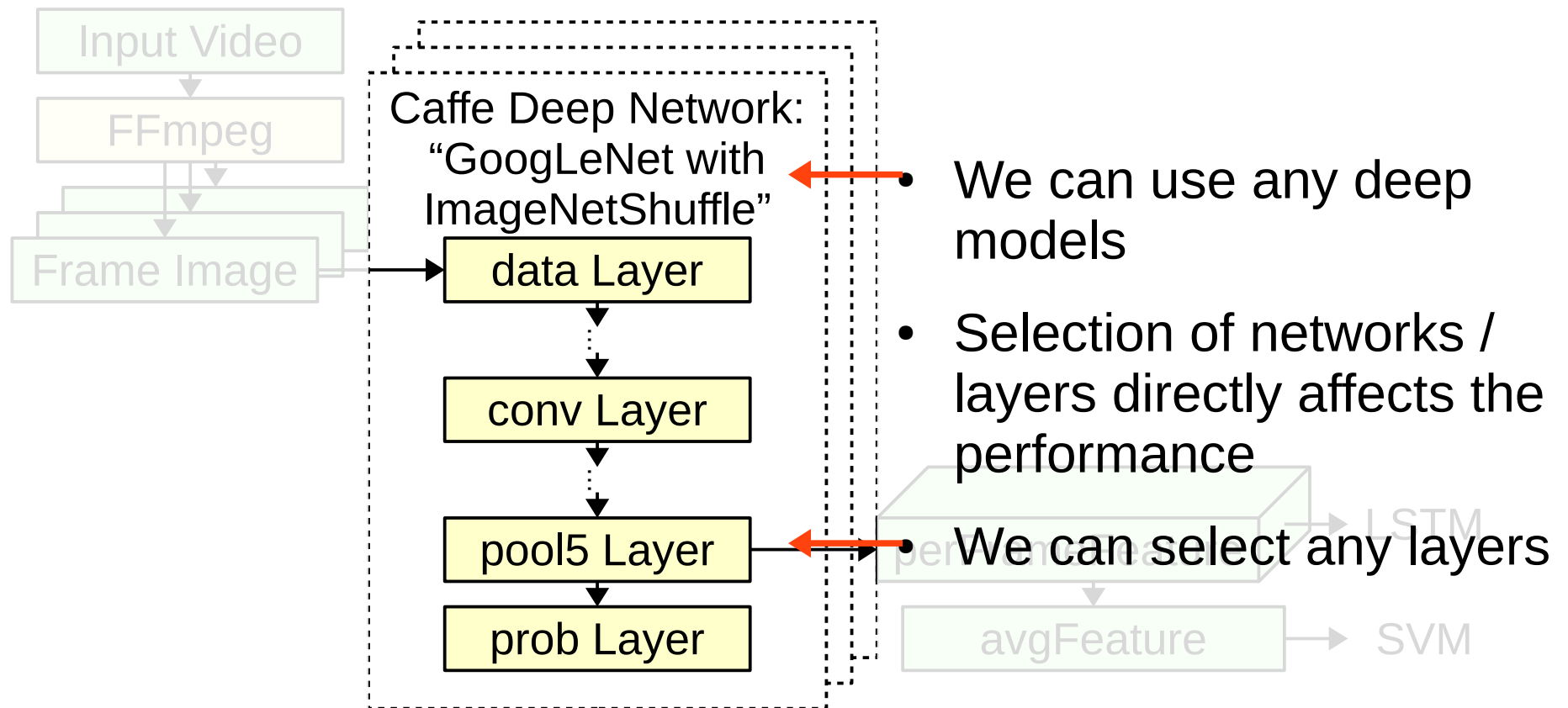
- Extract feature step overview

Extract feature: Overview

- Extract feature step overview

Input Video

FFmpeg

Frame Image

Caffe Deep Network: "GoogLeNet with ImageNetShuffle"

data Layer

conv Layer

pool5 Layer

prob Layer

per frame feature

avgFeature

LSTM

SVM

- We use a pre-trained deep network to extract feature

- Some of the layers are used as "feature vectors"

- We do not train the network due to a small number of training samples (MED)

Extract feature: Overview

- Extract feature step overview

- We can use any deep models

- Selection of networks / layers directly affects the performance

- We can select any layers

Extract feature: Requirements

- FFmpeg —— Built by us

- Caffe

- Python 2.7

  - numpy

  - h5py   Pre-installed on TSUBAME 3.0

  - scikit-image

```
> module load python-extension/2.7
```

  - sk-video —— Installed in a job script

```
> pip install --user sk-video
```

Extract feature: *Exercise*

- Run the actual feature extraction script

- You can edit the script by one of the following:

```
> cd MED-Task-master/extractFeature/
> gedit extractDeepFeaturesStarterFromVideo.sh
> emacs -nw extractDeepFeaturesStarterFromVideo.sh
> emacs extractDeepFeaturesStarterFromVideo.sh
```

- You can a job by the following:

```
> qsub -g tga-crest-deep extractDeepFeaturesStarterFromVideo.sh
```

- You can check the progress of the job by the following:

```
> qstat
> less eval.*
```

- You will get AP 0.931 for E023, 0.731 for E027

Extract feature: *Exercise*

- SVM is used for evaluation

- Run the SVM train / test and evaluation script

```
> cd ../svm/
> qsub -g tga-crest-deep svm.sh
```

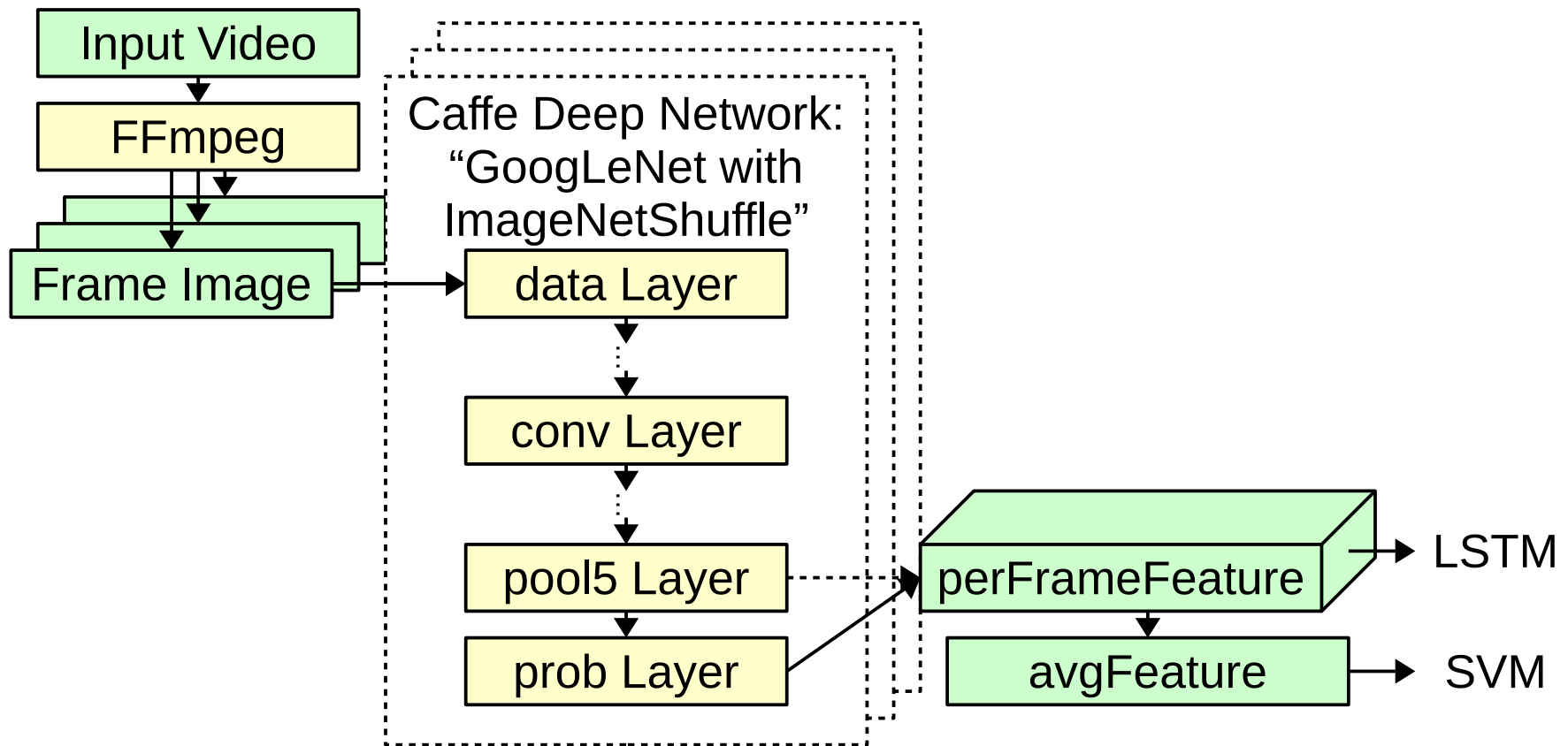- You can check the progress of the job by the following:

```
> qstat
> less feature.*
```

- You can check the result by:

```
> less TokyoTech_MED16_KINDRED_PS_100Ex_SML_DCNN-pool5-
STRICTREF_1/ap.csv
```

Extract feature: Modify

- We can select another layer to extract feature

Extract feature: *Exercise*

- Edit the actual feature extraction script to extract feature from a different layer

- You can edit the script by one of the following:

```
> cd ../extractFeature/
> gedit extractDeepFeaturesStarterFromVideo.sh
> emacs -nw extractDeepFeaturesStarterFromVideo.sh
> emacs extractDeepFeaturesStarterFromVideo.sh
```

- Edit line #17 as follows to switch to prob layer:

```
  @@ -17,1 +17,1 @@
- LAYER_NAMES=pool5/7x7_s1
+ LAYER_NAMES=prob
```

- Run

```
> qsub -g tga-crest-deep extractDeepFeaturesStarterFromVideo.sh
```

Extract feature: *Exercise*

- Edit the SVM script to use features from the different layer

- You can modify the script by one of the following:

```
> cd ../svm/
> sed -e 's/pool5/prob/g' svm.sh > svm_prob.sh
```

- Run

```
> qsub -g tga-crest-deep svm_prob.sh
```

- Check

```
> less TokyoTech_MED16_KINDRED_PS_100Ex_SML_DCNN-prob-
STRICTREF_1/ap.csv
```

- How is the result?

Agenda

4. How to train and test LSTM with Torch

- – Construct an LSTM network with Torch

- – Extend the network to Bi-LSTM

- – Extend the network to two layers

- – Train the network

- – Evaluate performance of the network

LSTM