TTI Prediction in Urban Road Network Using Computer Vision Techniques

LI Xiahan

Phase I: Feature Engineering

- Input: Transport Picture with size (768,896,3)

- Now still producing pictures

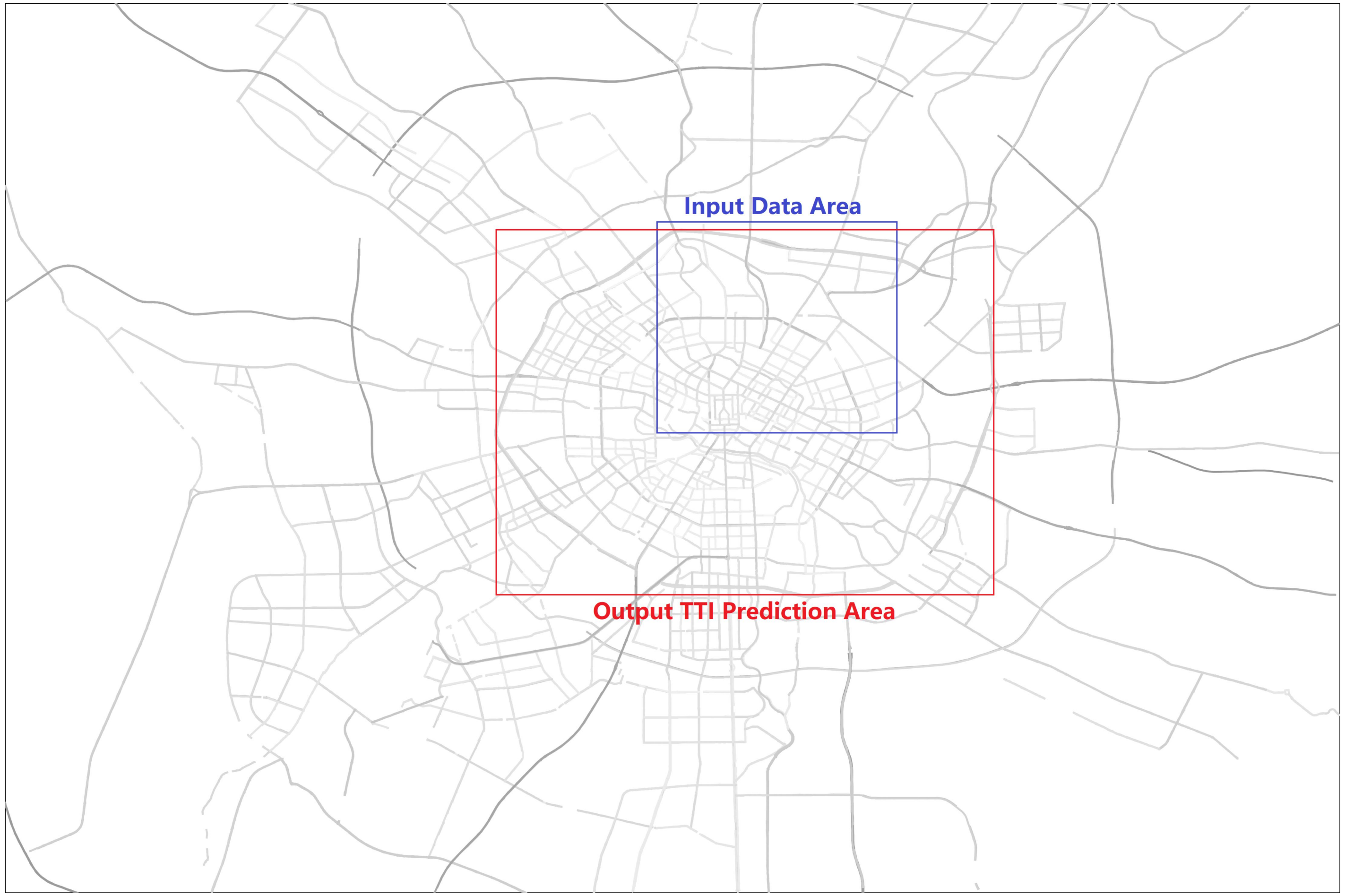
- Completed:
- 3000 pictures (covering 20 days)

 (1/3 of total)

Phase I: Feature Engineering

- Output: Overall TTI in a much bigger area.

1. When only limited data from part of city is available, we can still predict the transportation status of most urban area.
2. Increase the difficulties of prediction.
3. Prediction Area includes '3rd ring', where most of urban transportation happens



Phase I: Feature Engineering

Input & Output of Neutral Network in Phase II

Input Data Area:

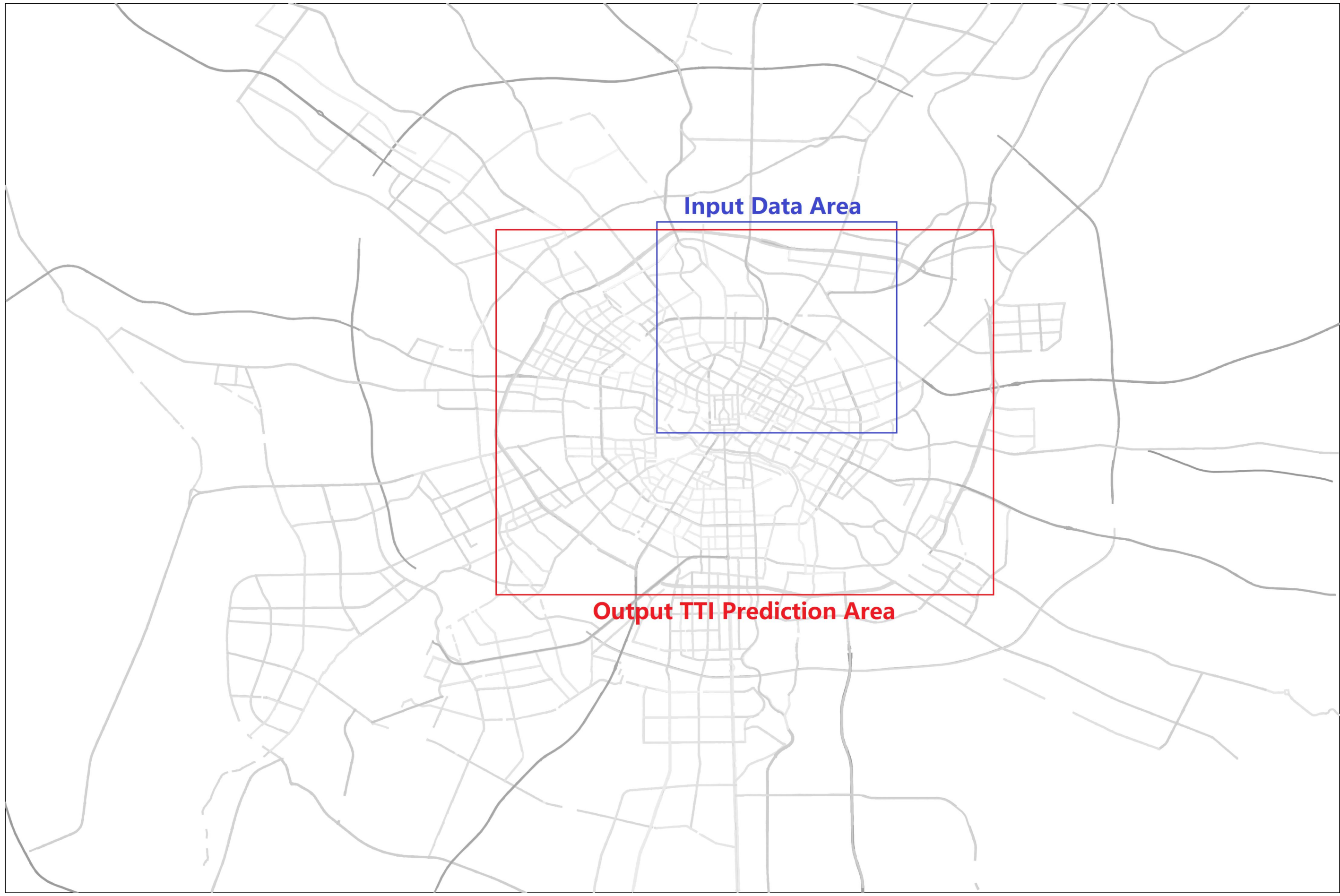
Longitude[104.0402, 104.1298]

Latitude [30.6516, 30.7284]

Output TTI Prediction Area:

Longitude[103.983681, 104.162492]

Latitude [30.594449, 30.726112]

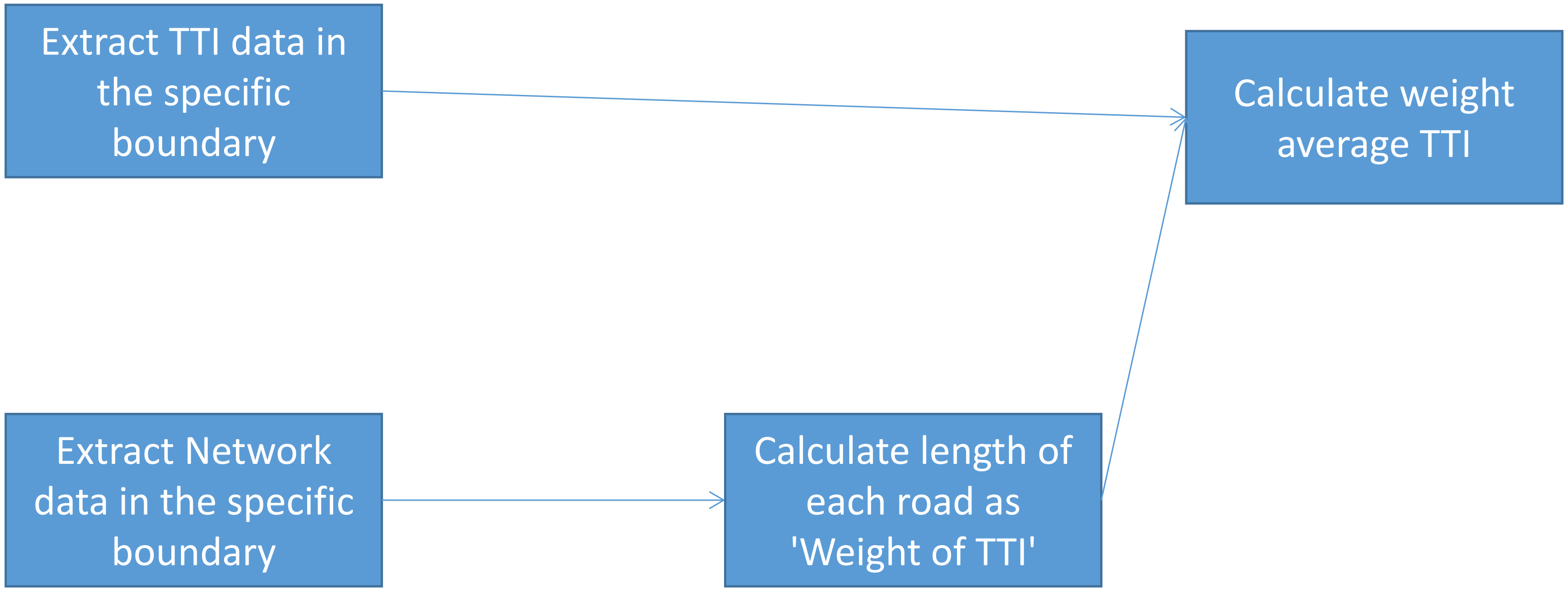


Generate output data

- Extract all the segments of roads in the TTI prediction area.
- Calculate the length of segments in each road as 'weight of TTI'.
- Extract TTI data from raw data.
- Group TTI data by the same time. For example, group all TTI data from 1859 roads at 2018-10-10 00:00:00 together.
- Based on 'weight of TTI', calculate the weight average of one timestamp.
- Do the previous grouping and calculating for every timestamp with 10min interval.
- At last, we get the list of weight average TTI in the specific area at every timestamp.

Generate output data

282659	2018/10/19 20:00	1.89428	21.0006
282964	2018/10/19 20:00	1.47032	24.84
281952	2018/10/19 20:00	1.35722	43.2719
281933	2018/10/19 20:00	1.50704	29.3833
282289	2018/10/19 20:00	2.61735	14.4739
282860	2018/10/19 20:00	1.38475	29.7054
282966	2018/10/19 20:00	1.08128	96.1934



181581	双流区	POLYGON((104.11863 30.22707,104.11661 30.22718,104.11644 30.22719,104.11491 3
181582	简阳市	POLYGON((104.42813 30.08827,104.42665 30.08851,104.42629 30.08857,104.42363 3
281863	八里桥路:北站东二路,三环路	MULTILINESTRING((104.07437 30.71442,104.0742 30.71483,104
281864	八里桥路:三环路,北站东二路	MULTILINESTRING((104.07452 30.71324,104.0745 30.71338,104
281865	二环路西段:广福路,金牛大道	MULTILINESTRING((104.02746 30.64001,104.02707 30.64039),(
281866	二环路西段:金牛大道,广福路	MULTILINESTRING((104.02536 30.64168,104.02488 30.64216,10

	A	B
1	DateTime	TTI
2	2018/10/10 0:00	1.1850004
3	2018/10/10 0:10	1.1898776
4	2018/10/10 0:20	1.1643739
5	2018/10/10 0:30	1.170975
6	2018/10/10 0:40	1.1708319
7	2018/10/10 0:50	1.1781483
8	2018/10/10 1:00	1.1747198
9	2018/10/10 1:10	1.1716122
10	2018/10/10 1:20	1.1742418
11	2018/10/10 1:30	1.1671112
12	2018/10/10 1:40	1.161023
13	2018/10/10 1:50	1.1865648
14	2018/10/10 2:00	1.1491824
15	2018/10/10 2:10	1.1543286
16	2018/10/10 2:20	1.1356649
17	2018/10/10 2:30	1.1309528
18	2018/10/10 2:40	1.1421093
19	2018/10/10 2:50	1.1469242
20	2018/10/10 3:00	1.15543
21	2018/10/10 3:10	1.1623394
22	2018/10/10 3:20	1.1579517
23	2018/10/10 3:30	1.1640627
24	2018/10/10 3:40	1.1627968
25	2018/10/10 3:50	1.1880888
26	2018/10/10 4:00	1.1573468
27	2018/10/10 4:10	1.1619954
28	2018/10/10 4:20	1.1610826
29	2018/10/10 4:30	1.150925
30	2018/10/10 4:40	1.1478054

Approximation Prove

In the same link in a time slice, tt_i = free flow speed / actual speed

In a collection of links s, $s = \{Link_1, Link_2, \cdots, Link_N\}$

size of s is N freeflow speed is $V_{free\text{-}i}$

Length of link is L_i realtime speed is V_i

Weight of link is W_i

$$TTI = \frac{\sum_{i=1}^{N} \frac{L_i}{V_i} \cdot W_i}{\sum_{i=1}^{N} \frac{L_i}{V_{free\text{-}i}} \cdot W_i}$$

In an area, we have $A = \{S_1, S_2, S_3, \cdots, S_p\}$. size of A is p.

Where S_k has size N_k. Based on definition,:

$$TTI_{weight} = \frac{\sum_{i=1}^{N_1} \frac{L_i}{V_i} \cdot W_i + \sum_{j=1}^{N_2} \frac{L_j}{V_j} \cdot W_j + \cdots + \sum_{\lambda=1}^{N_p} \frac{L_\lambda}{V_\lambda} W_\lambda}{\underbrace{\sum_{i=1}^{N_1} \frac{L_i}{V_{free\text{-}i}} W_i}_{S_1\ part} + \underbrace{\sum_{j=1}^{N_2} \frac{L_j}{V_{free\text{-}j}} \cdot W_j}_{S_2\ part} + \cdots + \underbrace{\sum_{\lambda=1}^{N_p} \frac{L_\lambda}{V_{free\text{-}\lambda}} W_\lambda}_{S_3\ part}}$$

But Based on our calculation, we have:

$$TTI_{weight}' = \frac{\sum_{i=1}^{N_1} L_i \cdot TTI_1 + \sum_{j=1}^{N_2} \cdot TTI_2 \cdot L_j + \cdots + \sum_{\lambda=1}^{N_p} TTI_{N_p} \cdot L_\lambda}{\sum_{i=1}^{N_1} L_i + \sum_{j=1}^{N_2} L_j + \cdots + \sum_{\lambda=1}^{N_p} L_\lambda}$$

where
$$TTI_s = \frac{\sum_{a=1}^{N_s} \frac{L_a}{V_a} W_a}{\sum_{a=1}^{N_s} \frac{L_a}{V_{free\text{-}a}} W_a}$$

we need to prove:

$$\boxed{TTI_{weight} = TTI_{weight'}}$$

Approximation Prove

Assumption: $\boxed{W_i = 1 \quad \text{for all } i}$ Because In big scale, we have omit the width of the road, so weight of road will not be considered. Only consider length.

Assumption: $\boxed{V_{free\_i} = V_{free\_j} \text{ for all } i, j.}$ Because in urban road, policymakers have an identical ~~speed~~. maximum speed, like 50km/h, to regulate all the urban road vehicles,

$$\text{So } TTI_{weight} = \frac{\sum_{i=1}^{N_1}\frac{L_i}{V_i}w_i + \sum_{j=1}^{N_2}\frac{L_j}{V_j}w_j + \cdots + \sum_{\lambda=1}^{N_p}\frac{L_\lambda}{V_\lambda}w_\lambda}{\sum_{i=1}^{N_1}\frac{L_i}{V_{free\text{-}i}}w_i + \sum_{j=1}^{N_2}\frac{L_j}{V_{free\text{-}j}}w_j + \cdots + \sum_{\lambda=1}^{N_p}\frac{L_\lambda}{V_{free\text{-}\lambda}}w_\lambda}$$

$$\begin{pmatrix} W_i = 1 \\ V_{free\_i} = V_{free\text{-}j} \end{pmatrix} = \frac{\sum_{i=1}^{N_1}\frac{L_i}{V_i} + \sum_{j=1}^{N_2}\frac{L_j}{V_j} + \cdots + \sum_{\lambda=1}^{N_p}\frac{L_\lambda}{V_\lambda}}{\sum_{i=1}^{N_1}L_i + \sum_{j=1}^{N_2}L_j + \cdots + \sum_{\lambda=1}^{N_p}L_\lambda} \cdot V_{free}$$

$$= V_{free}$$

$$= \frac{1}{\sum_{i=1}^{N_1}L_i + \sum_{j=1}^{N_2}L_j + \cdots + \sum_{\lambda=1}^{N_p}L_\lambda} \cdot \left(\sum_{i=1}^{N_1}\frac{L_i}{V_i}\cdot\frac{1}{V_{free}} + \sum_{j=2}^{N_2}\frac{L_j}{V_j}\cdot\frac{1}{V_{free}} + \cdots + \sum_{\lambda=1}^{N_p}\frac{L_\lambda}{V_\lambda}\cdot\frac{1}{V_{free}}\right)$$

$$= \frac{1}{\sum_{i=1}^{N_1}L_i + \sum_{j=1}^{N_2}L_j + \cdots + \sum_{\lambda=1}^{N_p}L_\lambda} \cdot \left(\sum_{i=1}^{N_1}L_i\cdot\frac{\frac{L_i}{V_i}w_i}{\frac{L_i}{V_{free}}w_i} + \sum_{j=1}^{N_j}L_j\cdot\frac{\frac{L_j}{V_j}w_j}{\frac{L_j}{V_{free}}w_j} + \cdots\right)$$

$$= \frac{\sum_{i=1}^{N_1}TTI_1 L_i + \sum_{j=1}^{N_2}TTI_2 L_j + \cdots + \sum_{\lambda=1}^{N_p}TTI_1 L_\lambda}{\sum_{i=1}^{N_1}L_i + \sum_{j=1}^{N_2}L_j + \cdots + \sum_{\lambda=1}^{N_p}L_\lambda} = TTI_{weight'}$$

Phase II: Deep Learning

- In phase I, we get transportation pictures as input, and weight TTI in urban area as output.
- In phase II, we will apply different neural network to do deep learning, and check the validation loss of them, and comment the results.

Structure of Input

- Divide all pictures as inputs:

1. Training set(picture folder)
2. Validation set(testpic folder)

- Divide the weight TTI values into two csv files, as outputs:

1. Traning results(train.csv)
2. Validation results(test.csv)

Criteria(till now)

- Four validation tools:

1. Average TTI(Baseline)
2. CNN(Resnet34)
3. CNN(Resnet34)+LSTM
4. ST-Resnet

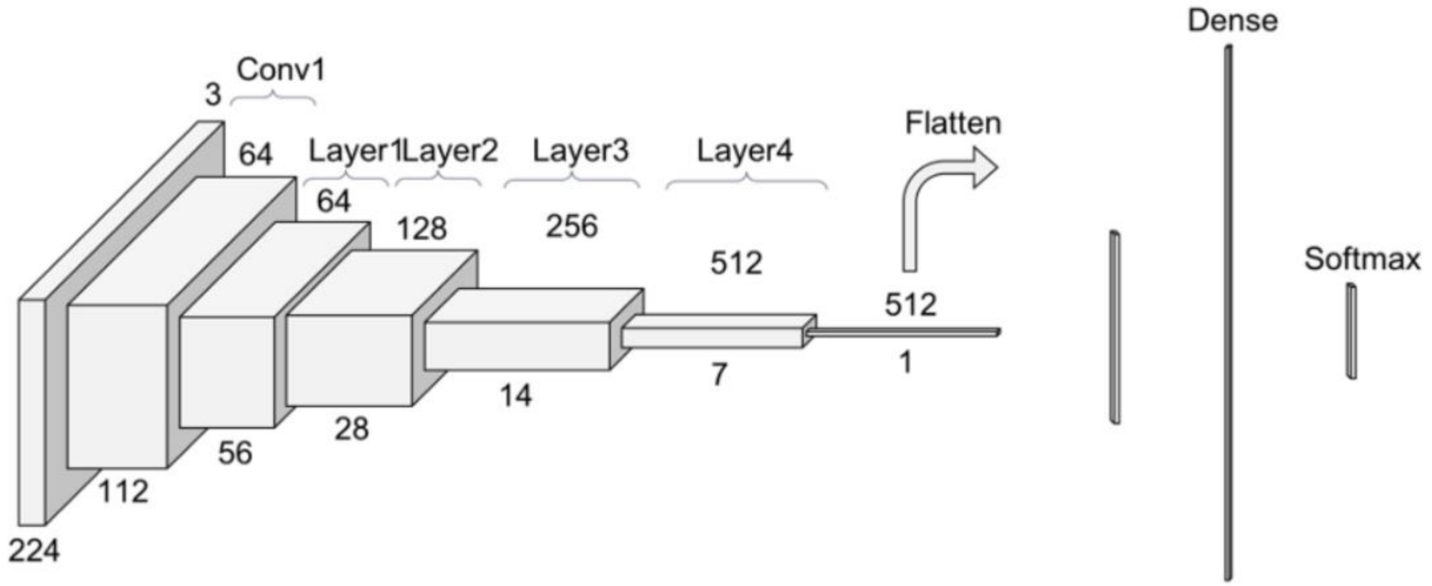
- Metrics: natural logarithm of L1 loss, ln(y')-ln(y)
- Inputs: 720 training pictures and 144 validation pictures.(Will be increased to about 6000 training pictures and about 720 validation pictures)
- Criteria: Minimum metrics of validation tools that can be reached

Average TTI

- Directly compute average TTI for the same timestamp from everyday.
- Example: To predict TTI value at 2018-11-25, 12:00:00, use the average TTI value of 12:00:00 from 2018-10-08 to 2018-11-20
- Performed as a baseline.

CNN(Resnet34)

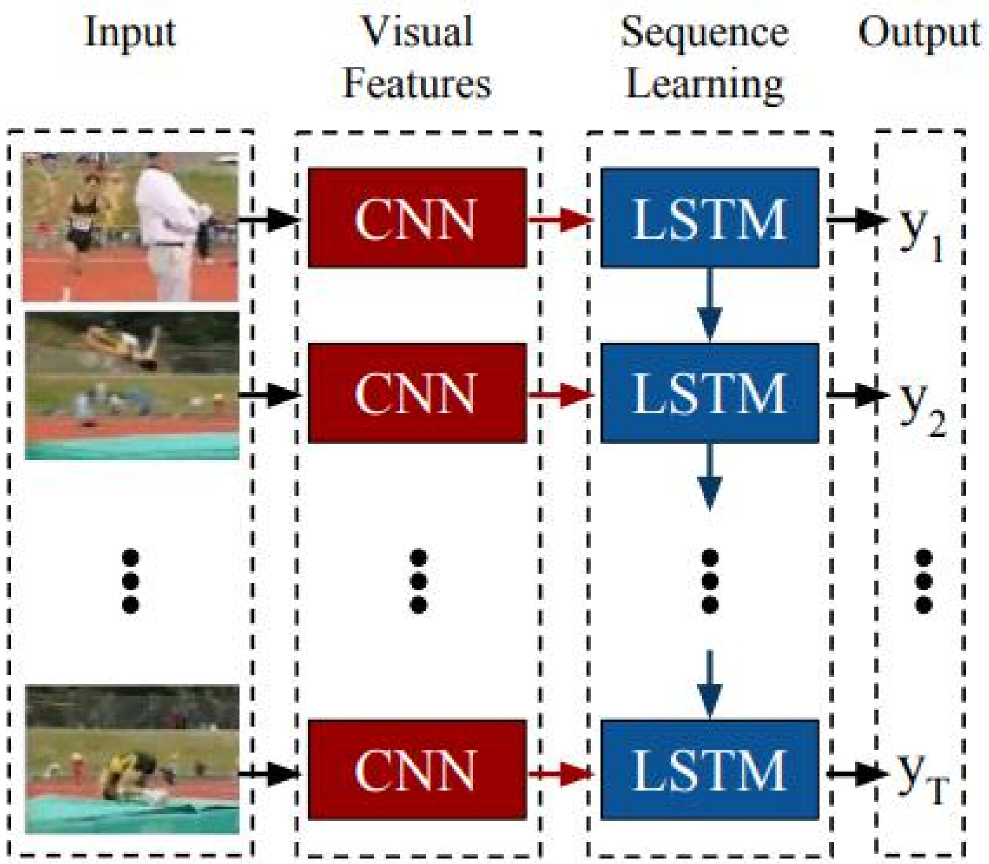
- Resnet is one kind of neural network raised in 2015.
- Now we have resnet18, resnet34, resnet50, resnet101, resnet152
- Perform as backbones in the deep learning model



https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8
https://arxiv.org/pdf/1512.03385.pdf

CNN(Resnet34)+LSTM

- LSTM could perform well in a time-related sequence of pictures
- Combine CNN and LSTM, we are expect to have a better performance than pure CNN.



https://arxiv.org/pdf/1411.4389.pdf

ST-Resnet

- In a transport prediction related paper, researchers use ST-Resnet to predict the time-related transport.
- Because the ST-Resnet seems to be highly customed in a specific data (NYC Bike and Beijing Taxi). Not sure what performance can it make

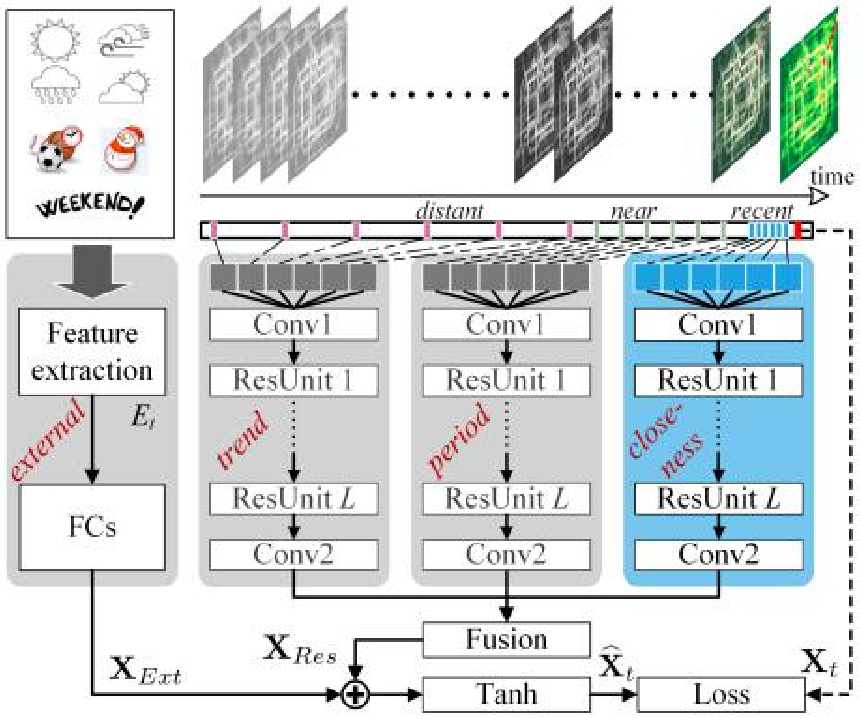


Figure 3: ST-ResNet architecture. Conv: Convolution; ResUnit: Residual Unit; FC: Fully-connected.

Future Plan

- Finish Phase II, at least get the results of 4 validation tools
- Proceeding pictures generation
- Master thesis writing
- Discuss about plan after master graduation. (PhD, go to Germany)