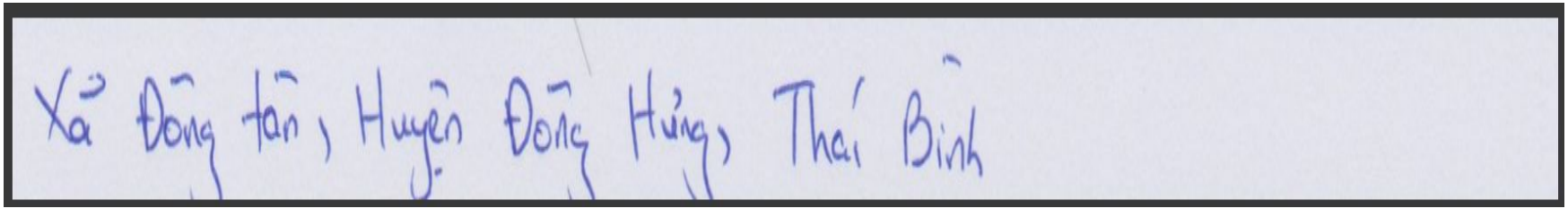
VIETNAMESE HANDWRITING RECOGNITION

Phan Quoc Khoi - ITDSIU18028

Index

- Why OCR
- Data
- CNN , RNN recap
- CRNN Model
- CTC (Connectionist Temporal Classification)
- Pipeline
- Application
- Ref

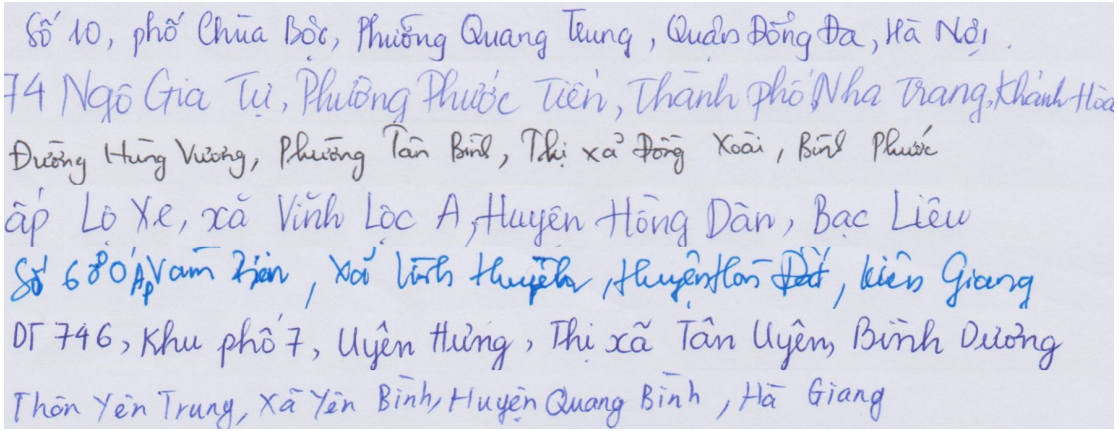
My Goal



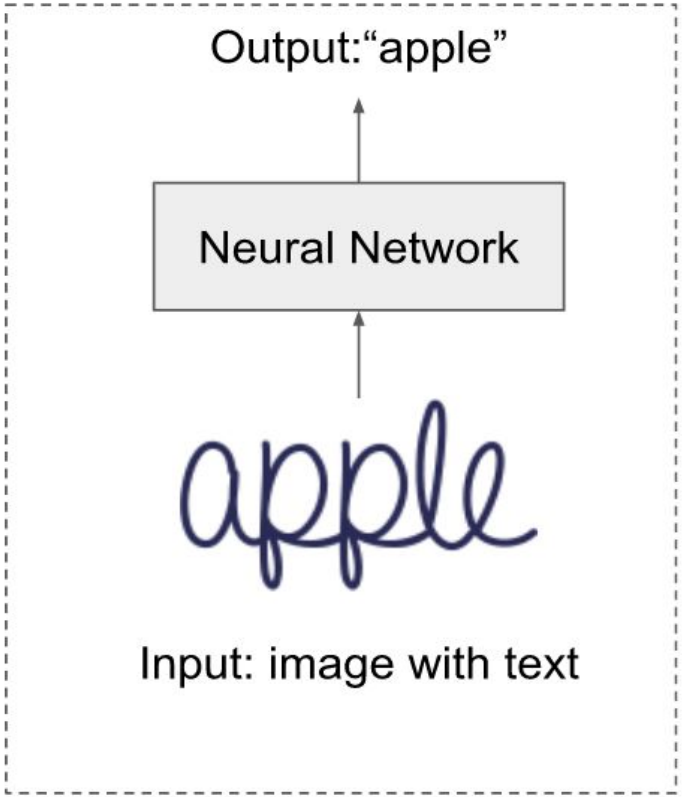
—> Xã Đông Tân, Huyện Đông Hưng, Thái Bình

Data

- Downloaded from [Cinnamon AI]
- Include 1838 images and 1 json label
- Size: 353MB




```
{'1.jpg': 'Số 3 Nguyễn Ngọc Vũ, Hà Nội',
 '2.jpg': 'Số 30 Nguyễn Hồng, Láng Hạ, Đống Đa, Hà Nội',
 '3.jpg': '58 Thái Thịnh, Đống Đa, Hà Nội',
 '4.jpeg': 'Số 370/8 khu phố 5B, phường Tân Biên, Biên Hòa, Đồng Nai',
 '5.jpg': 'Vĩnh Trung Plaza, B, 255-257 đường Hùng Vương, phường Vĩnh Trung',
 '6.jpg': 'Tòa nhà 34T, Hoàng Đạo Thúy, Hà Nội',
 '7.jpg': '40 Cát Linh, Đống Đa, Hà Nội',
 '8.jpg': 'phòng 101, tầng 1, lô 04-TT5B, khu đô thị Tây Nam Linh Đàm',
 '9.JPG': 'Nhà 87 ngõ 416 Đê La Thành',
 '10.JPG': 'Up coworking Space, 89 Láng Hạ, Hà Nội',
 '11.jpg': '192 Ngô Đức Kế, quận 1, Hồ Chí Minh',
 '12.jpg': 'số 5 Công Trường Mê Linh, phường Bến Nghé, quận 1',
 '13.jpg': '90A đường Mai Xuân Thưởng, tỉnh Gia Lai',
 '14.jpg': '96/7/12B Phạm Văn Đồng, thành phố Pleiku',
 '15.jpg': '168 Ngô Gia Tự, thành phố Hà Tĩnh',
 '0000_samples.png': 'Số 253 đường Trần Phú, Thị trấn Nam Sách, Huyện Nam Sách, Hải Dương',
 '0001_samples.png': 'Số 289 Đinh Bộ Lĩnh, Phường 26, Quận Bình Thạnh, TP Hồ Chí Minh',
 '0002_samples.png': 'Số 246E/2, Khu phố 1B, Phường An Phú, Tx Thuận An, Bình Dương',
 '0003_samples.png': '42 Đặng Thị Nhu, Phường Nguyễn Thái Bình, Quận 1, TP Hồ Chí Minh',
 '0004_samples.png': '200 Phan Bội Châu, Phường Trường An, Thành phố Huế, Thừa Thiên - Huế',
 '0005_samples.png': '27A Hoàng Việt, Phường 4, Quận Tân Bình, TP Hồ Chí Minh',
 '0006_samples.png': 'Số 113 Đường Hưng Thái 2, Phường Tân Phong, Quận 7, TP Hồ Chí Minh',
```

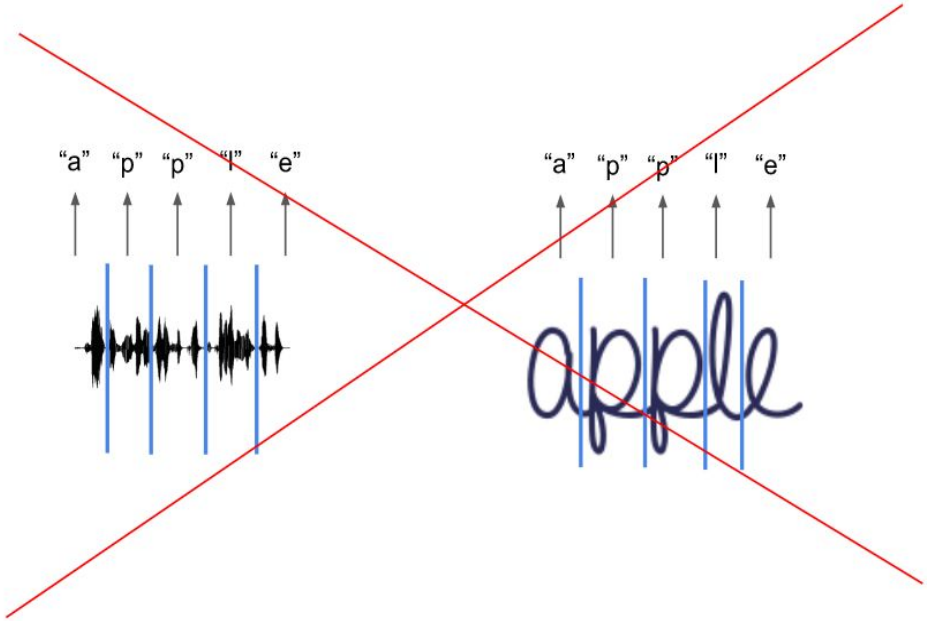


Example: image ocr

Challenges

We cannot pre-segment input data because:

- Impossible to do in most cases
- Expensive
- Time consuming

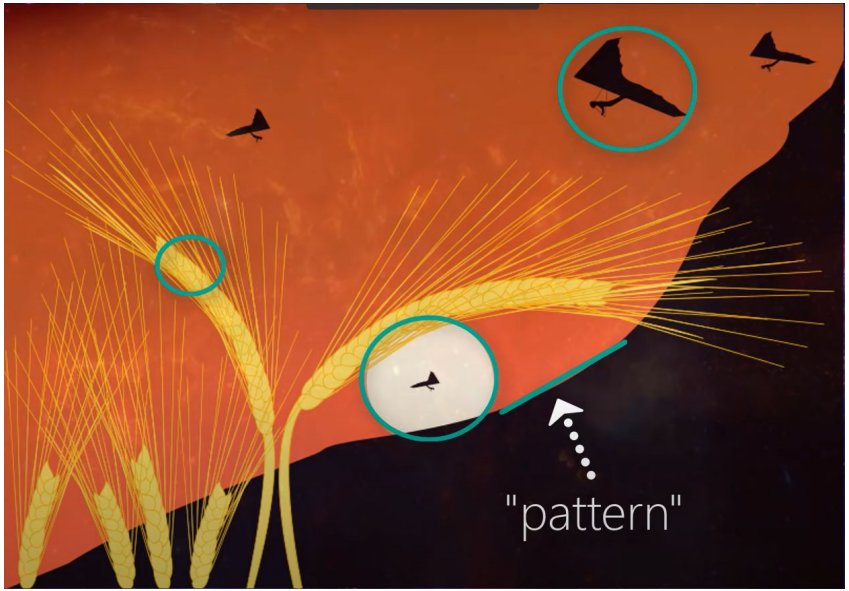


Solutions

- CRNN (CNN + RNN)
- Multiple Dimensions RNN (MDRNN)
- Scan, Attend and Read (SAR)
- Online, Offline OCR
- etc

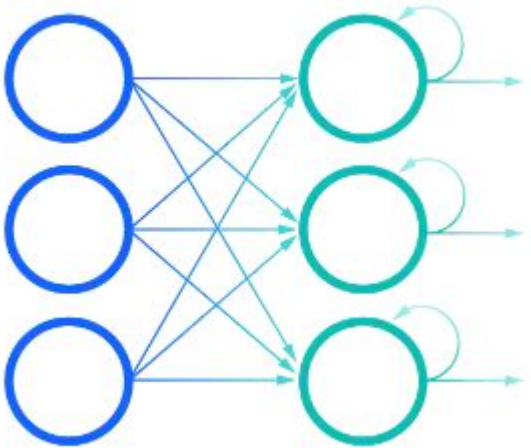
What is CNN (Convolutional Neural Network) ?

- An artificial network that have some type of specialization for being able to pick out or detect pattern from input and make sense of them



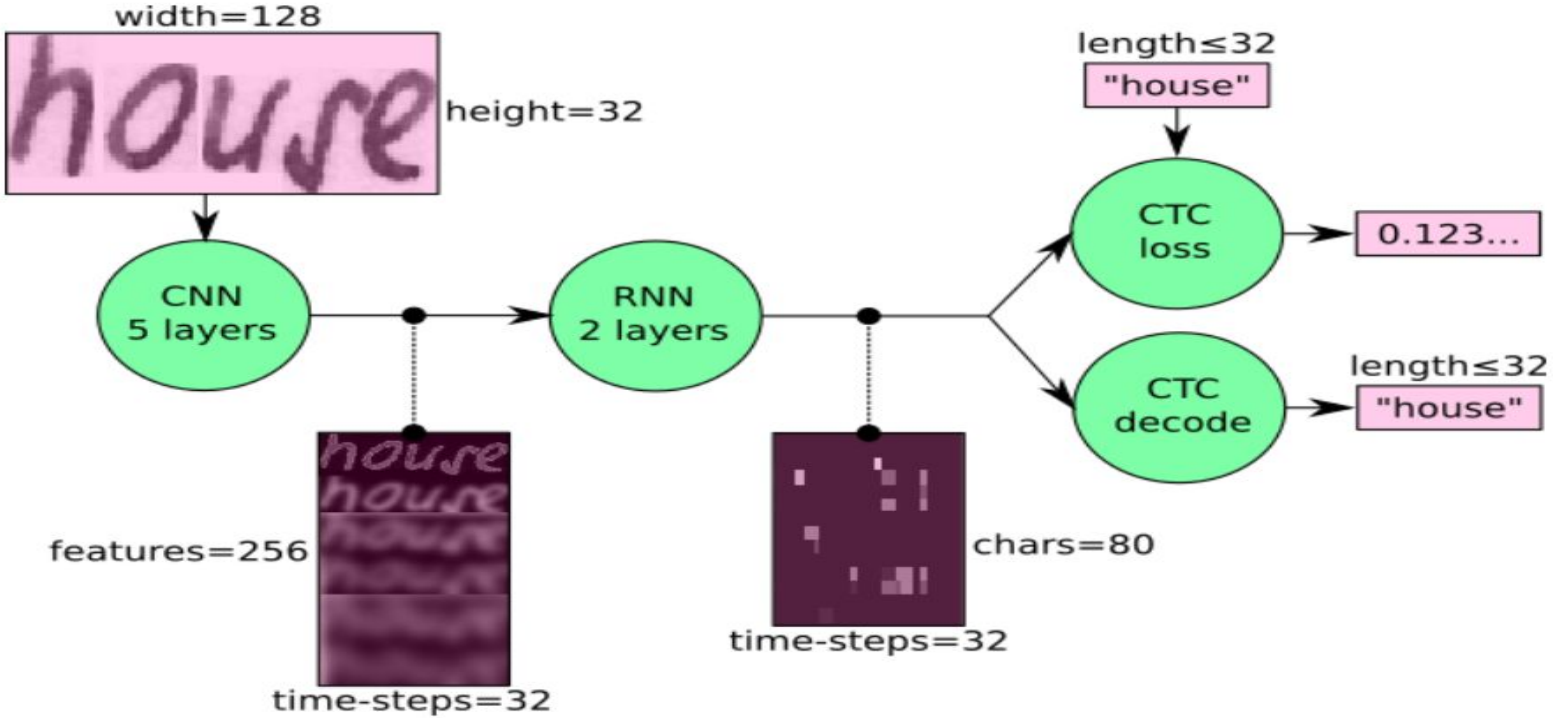
How about RNN (Recurrent Neural Network) ?

- An artificial neural network which uses sequential data or time series data. It is mostly used in NLP, language translation, image captioning, etc

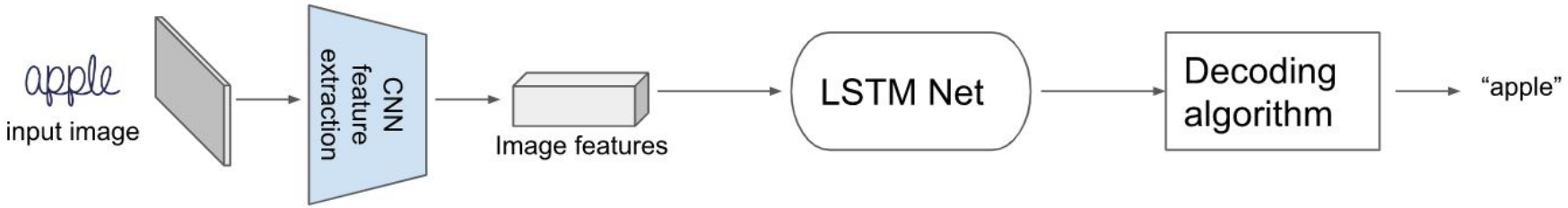


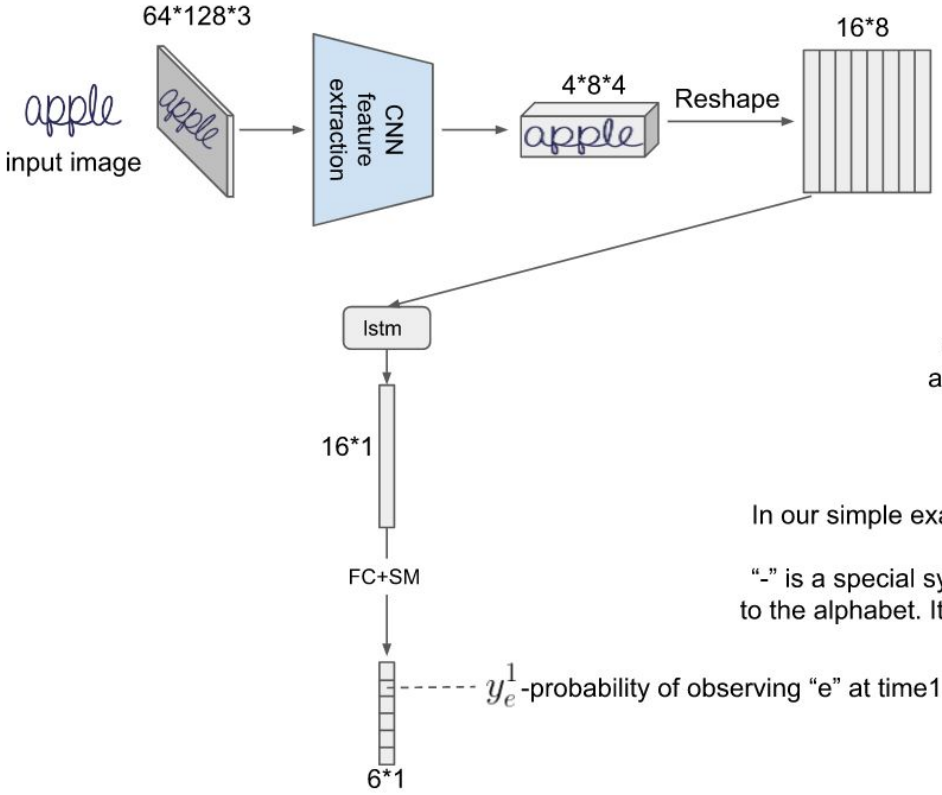

CNN
RNN
CRNN
imgflip.com
FUTUBANDERA.CL

CRNN Model Overview



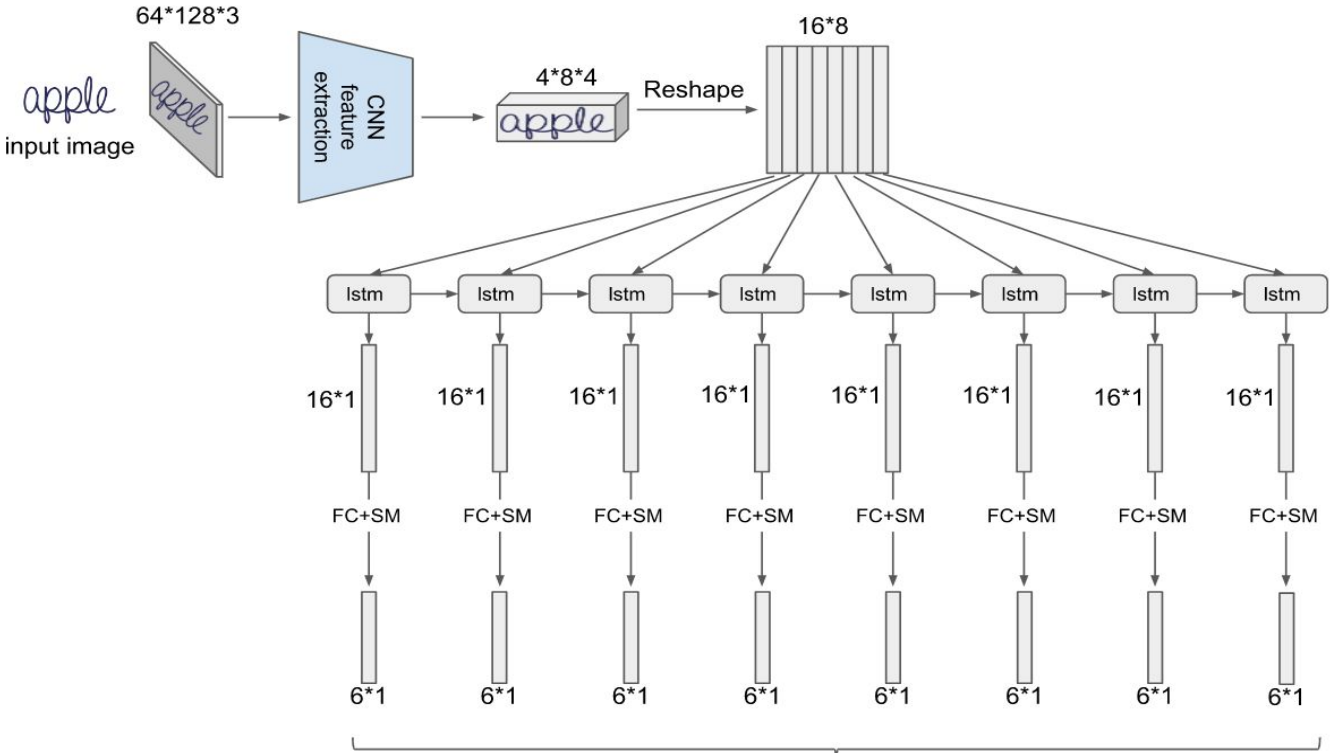
Explanation





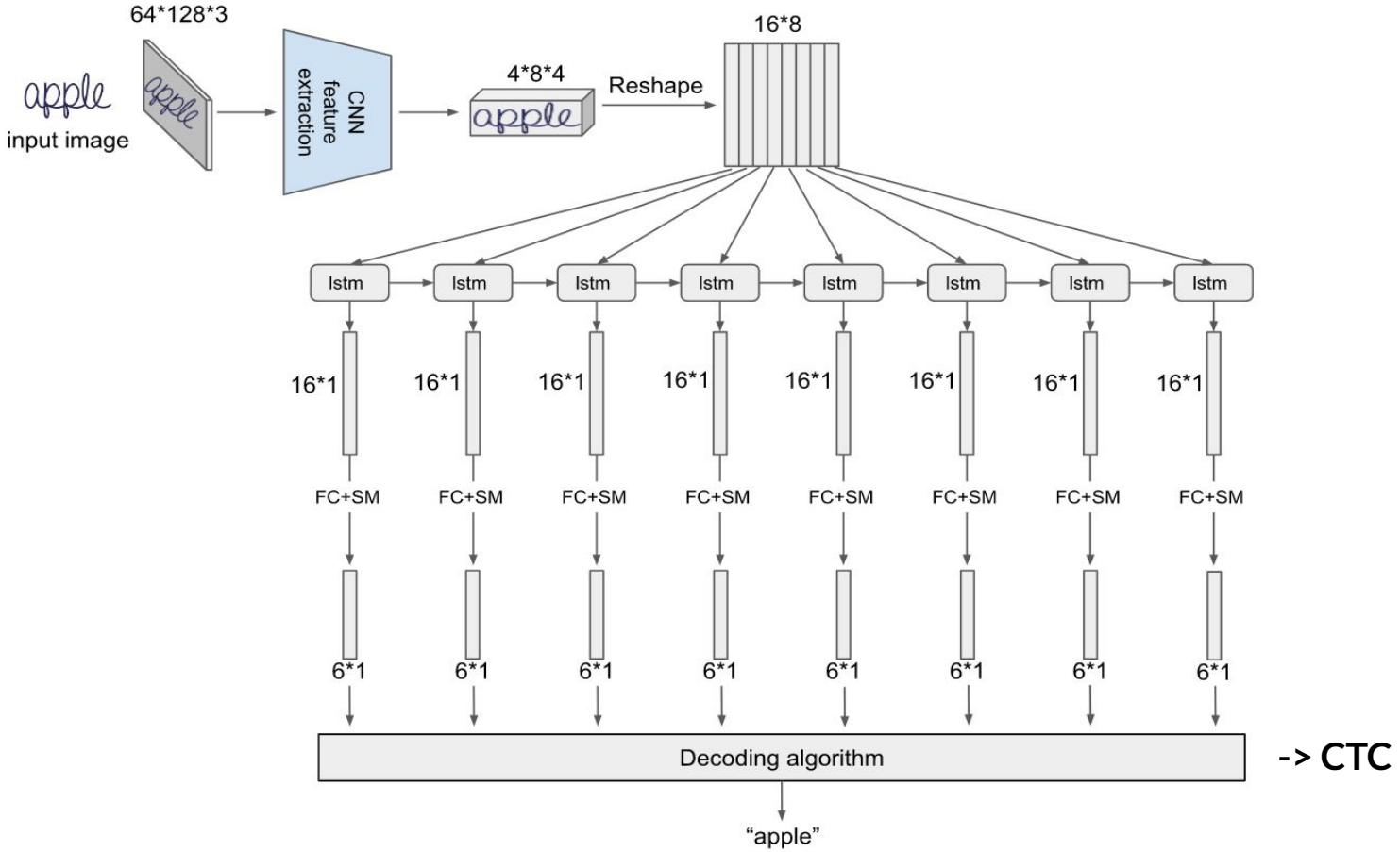
This is the probability distribution of observing alphabet symbols at time1 (time1 - first lstm step)

In our simple example: Alphabet={“a”, “e”, “l”, “p”, “z”, “-”}
|Alphabet| = 6
“-” is a special symbol (blank) that we always should add to the alphabet. It will be further understood what it is used for.



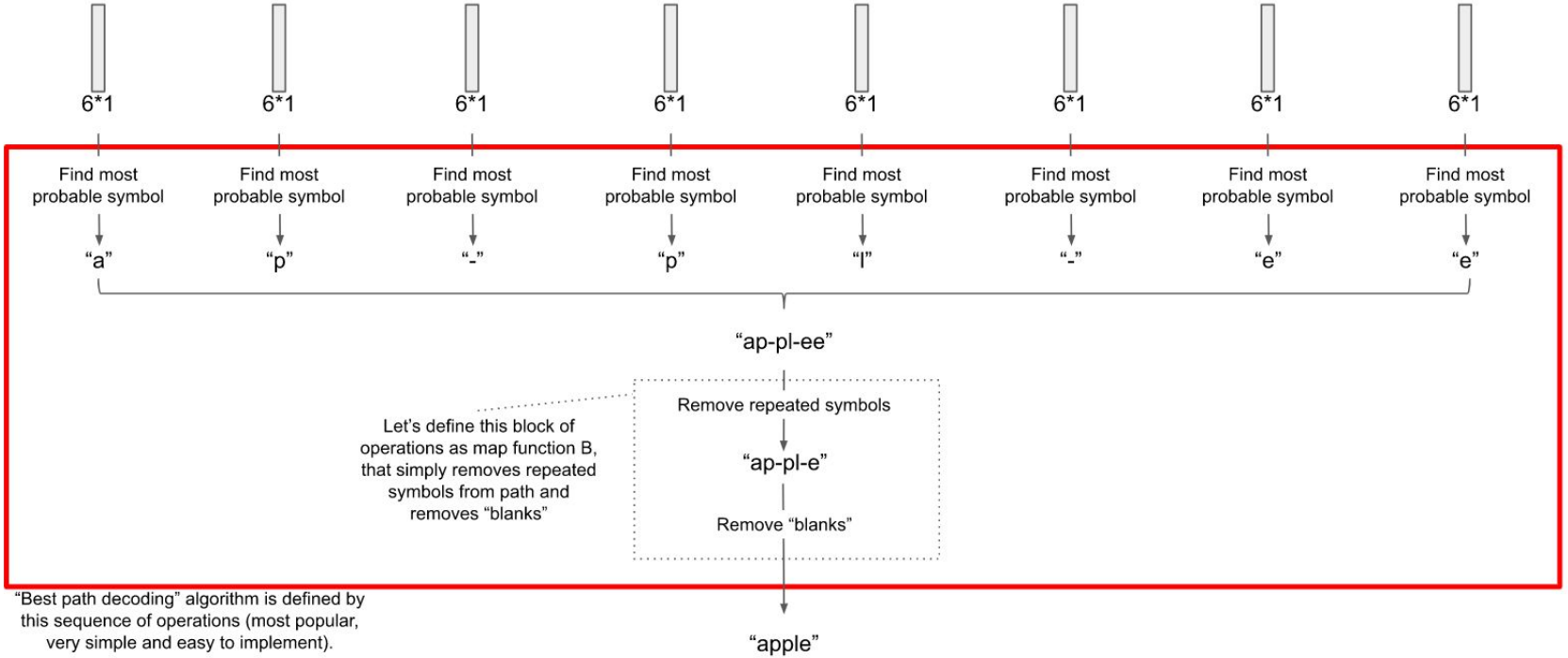
Note: We designed simplified neural network to have 8 outputs. It means that we can not recognize more than 8 characters per image.

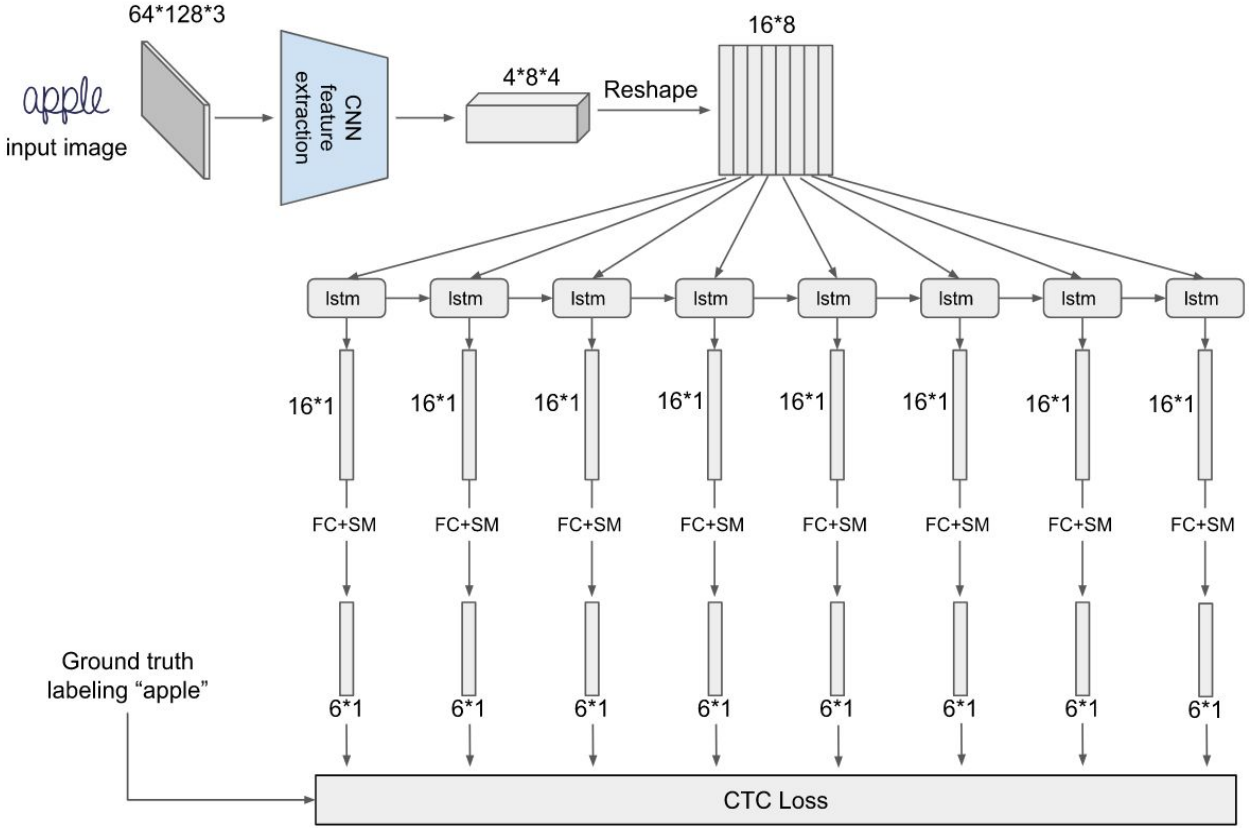
In practice, number of outputs can reach 32, 64 or more. The choice will depend on the specific task.


64*128*3
input image
CNN feature extraction
4*8*4
Reshape
16*8
lstm
16*1
FC+SM
6*1
Decoding algorithm
"apple"

-> CTC

Decode: Connectionist Temporal Classification (CTC)





Now we can train this model using CTC Loss

CTC LOSS

Path1: “ap-pl-ee” —— B(“ap-pl-ee”) ——→ Labeling: “apple”

$$p(\text{“}ap-pl-ee\text{”}) = y_a^1 \cdot y_p^2 \cdot y_-^3 \cdot y_p^4 \cdot y_l^5 \cdot y_-^6 \cdot y_e^7 \cdot y_e^8$$

Path1: “ap-pl-ee” —B(“ap-pl-ee”)→ Labeling: “apple”

$$p(\text{“}ap-pl-ee\text{”}) = y_a^1 \cdot y_p^2 \cdot y_-^3 \cdot y_p^4 \cdot y_l^5 \cdot y_-^6 \cdot y_e^7 \cdot y_e^8$$

Path2: “aapp--le” —B(“aapp--le”)→ Labeling: “aple”

$$p(\text{“}aapp--le\text{”}) = y_a^1 \cdot y_a^2 \cdot y_p^3 \cdot y_p^4 \cdot y_-^5 \cdot y_-^6 \cdot y_l^7 \cdot y_e^8$$

Path1: “ap-pl-ee” —B(“ap-pl-ee”)→ Labeling: “apple”

$$p(\text{“}ap-pl-ee\text{”}) = y_a^1 \cdot y_p^2 \cdot y_-^3 \cdot y_p^4 \cdot y_l^5 \cdot y_-^6 \cdot y_e^7 \cdot y_e^8$$

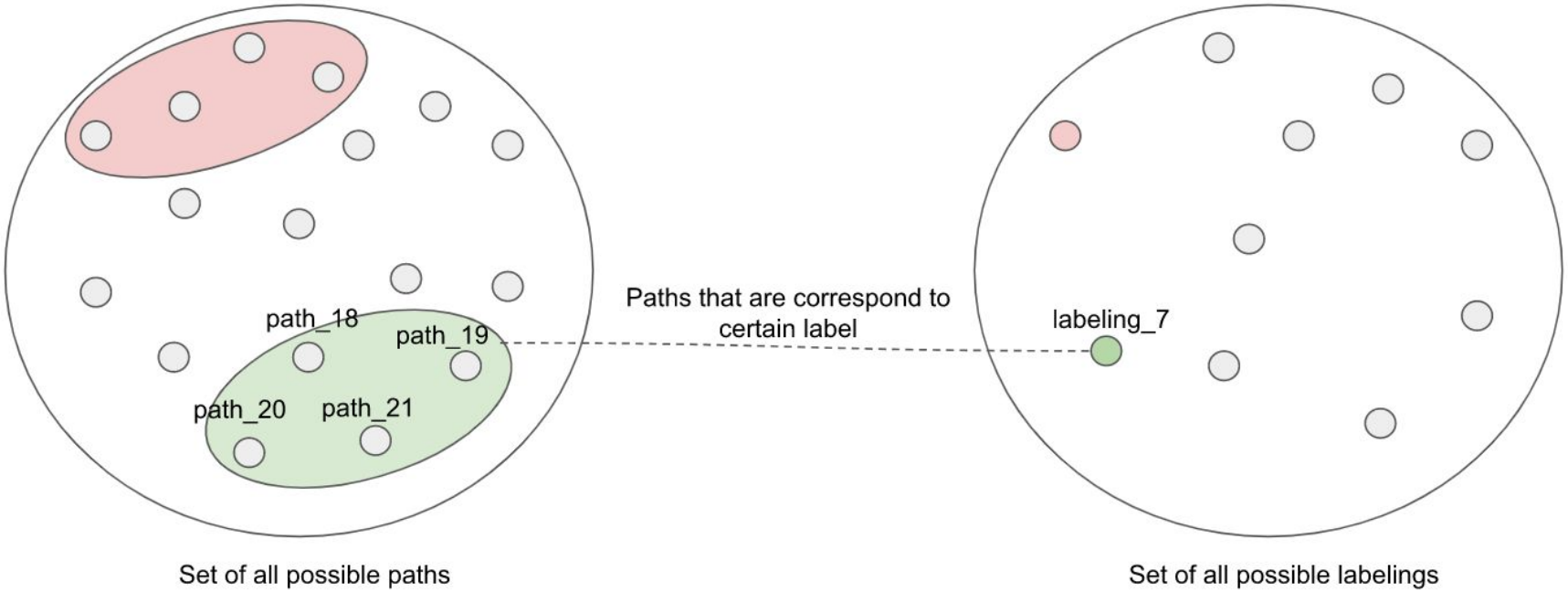
Path2: “aapp--le” —B(“aapp--le”)→ Labeling: “aple”

$$p(\text{“}aapp--le\text{”}) = y_a^1 \cdot y_a^2 \cdot y_p^3 \cdot y_p^4 \cdot y_-^5 \cdot y_-^6 \cdot y_l^7 \cdot y_e^8$$

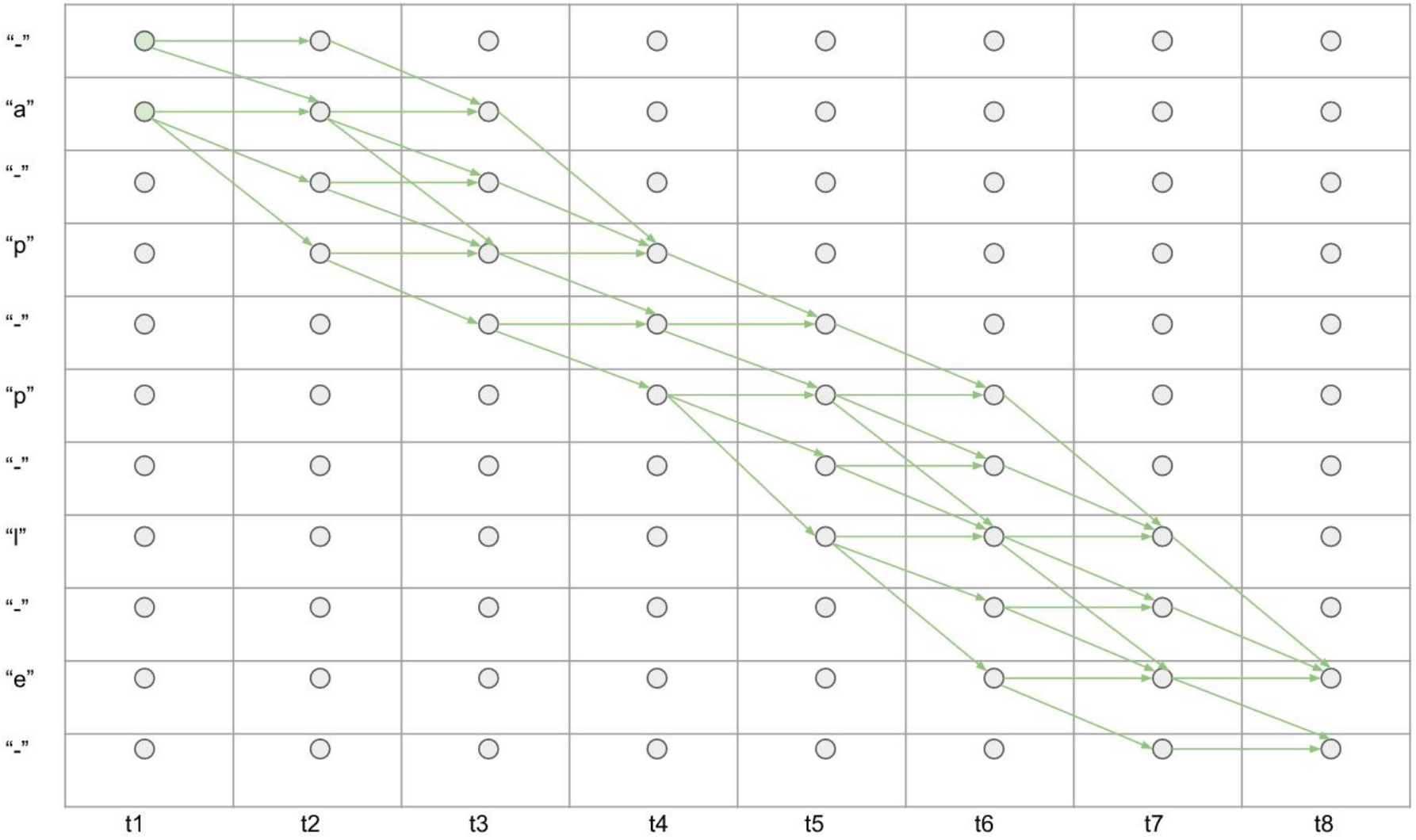
Path3: “aap--lzz” —B(“aap--lzz”)→ Labeling: “aplz”

... ...

PathN: “--z--eee” —B(“--z--eee”)→ Labeling: “ze”



p(labeling_7) = sum of probabilities of all corresponding paths =
= p(path_18) + p(path_19) + p(path_20) + p(path_21)


“-”
“a”
“-”
“p”
“-”
“p”
“-”
“l”
“-”
“e”
“-”
t1
t2
t3
t4
t5
t6
t7
t8

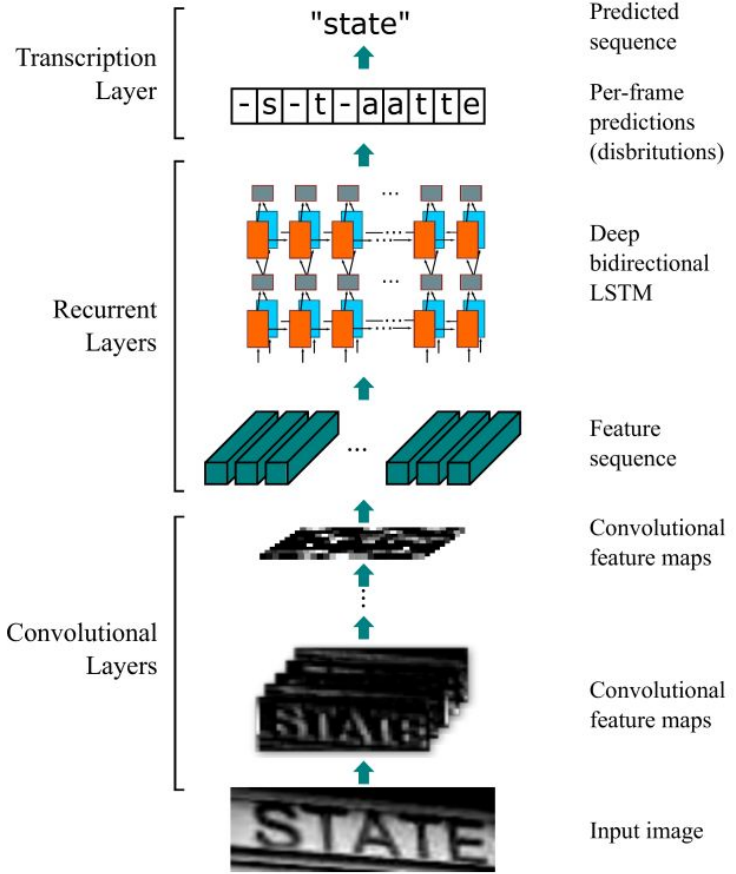
Another possible path examples

- gun -> guun, -gun, g-un, gun- => p(gun) = p(guun) + p(-gun) + p(g-un) + p(gun-)
- sun -> ssun, suun, su-n => p(sun) = p(ssun) + p(suun) + p(su-n)
- snow -> s–now, ssnow, snoow, sno-w => p(snow) = p(s-now) + p(ssnow) + p(snoow) + p(sno-w)

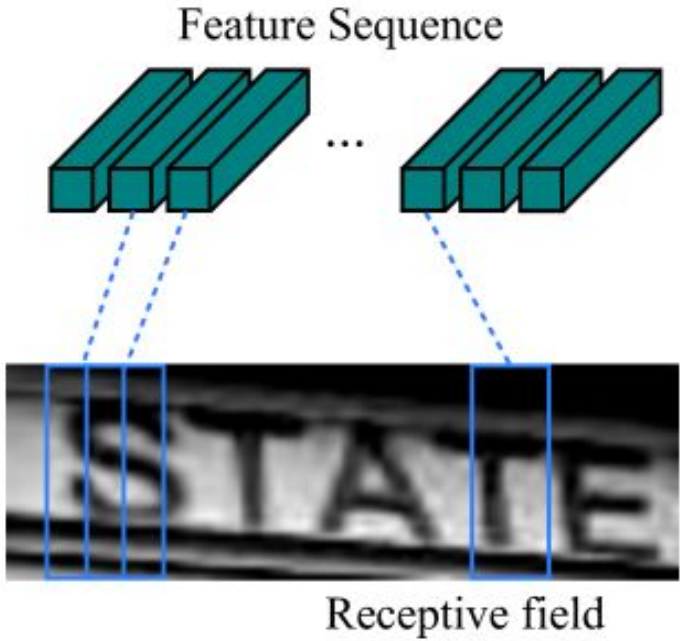
CRNN Structure

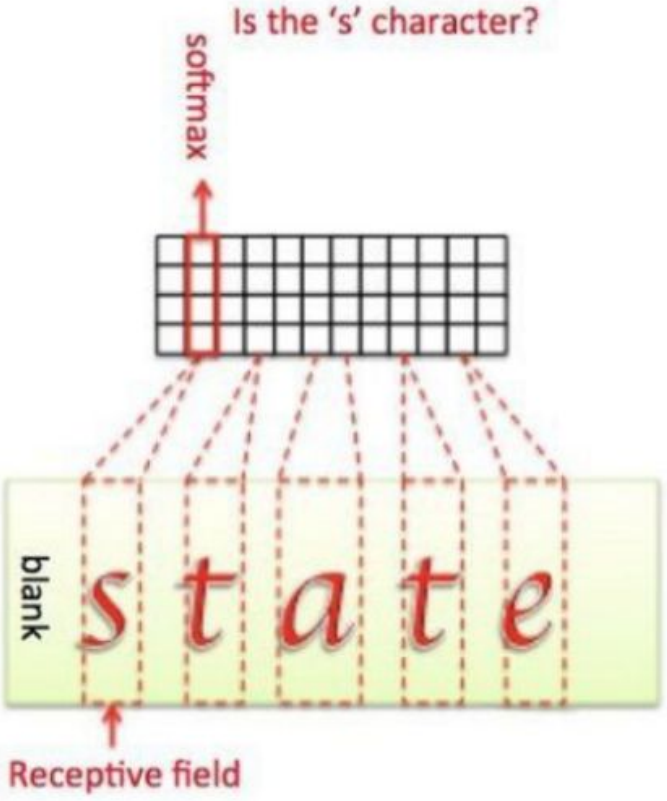
7 layers in total:

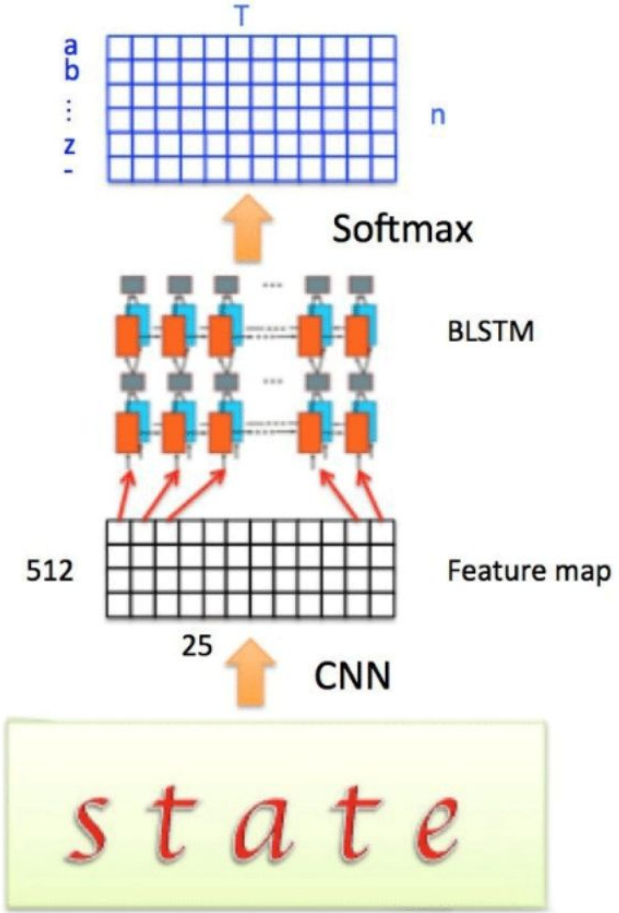
- 5 CNN: extract feature sequence
- 2 RNN: predict label distribution for each frame
- Transcription layer: translate per-frame prediction into final label sequence



Each column of the feature maps corresponds to a rectangle region of the original image (termed the *receptive field*)




Is the 's' character?
softmax
blank
s t a t e
Receptive field


T
a
b
z
-
n
Softmax
BLSTM
512
Feature map
25
CNN
s t a t e

Model Summary

- The 1st row is the top layer
- ‘k’, ‘s’, ‘p’ stand for kernel size, stride, padding size respectively

Type	Configurations
Transcription	-
Bidirectional-LSTM	#hidden units:256
Bidirectional-LSTM	#hidden units:256
Map-to-Sequence	-
Convolution	#maps:512, k:2×2, s:1, p:0
MaxPooling	Window:1×2, s:2
BatchNormalization	-
Convolution	#maps:512, k:3×3, s:1, p:1
BatchNormalization	-
Convolution	#maps:512, k:3×3, s:1, p:1
MaxPooling	Window:1×2, s:2
Convolution	#maps:256, k:3×3, s:1, p:1
Convolution	#maps:256, k:3×3, s:1, p:1
MaxPooling	Window:2×2, s:2
Convolution	#maps:128, k:3×3, s:1, p:1
MaxPooling	Window:2×2, s:2
Convolution	#maps:64, k:3×3, s:1, p:1
Input	$W \times 32$ gray-scale image

Data Preprocessing

- Find image width and height for scaling
- Read image using cv2
- Convert image (greyscale, blur, threshold)
- Resize image to same size for model input
- Split data into train test with 80-20 ratio

Result - Evaluation

It tooks me 3 hours to train the model with :

- Character Error Rate: 0.039
- Word Error Rate: 0.124
- Sequence Error Rate: 0.6766

```
Result:
 CER: 0.038720876512948955
 WER: 0.12437966090459698
 SER: 0.6766304347826086
```

OCR Demo

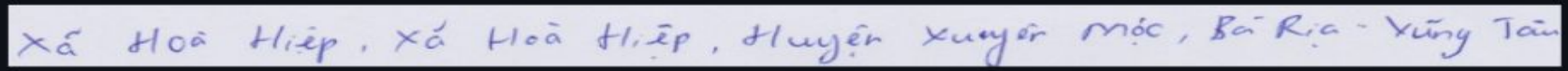
Select file

Drag and drop file here
Limit 200MB per file

Browse files

demo2.png 190.9KB

Your img



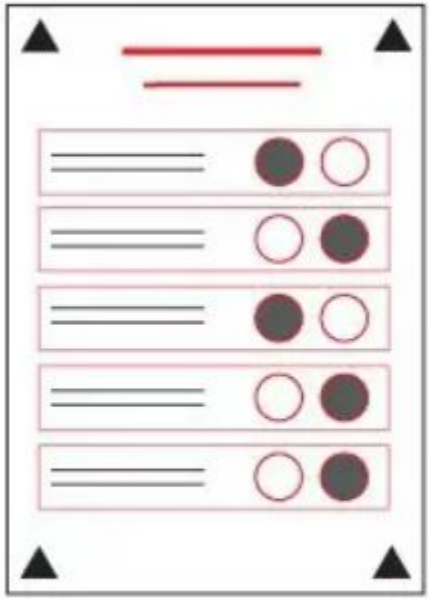
After preprocessing

Predict string:

Xã Hòa Hiệp, Xã Ma Hiệp, Huyện Xuyên Mộc, Bà Rịa Vũng Tà

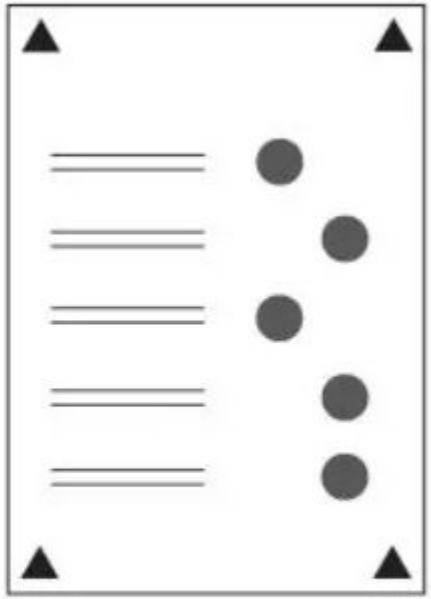
Application

- Insurance
- Hospital billing
- Post office
- Auto marking in education



Completed Form

Scan
(with Drop Out Colour Enabled)



Scanned Image

LET'S DEMO

Summary

Futher developement:

- More data
- Add more layers to model
- User input for real time predict - Online OCR

Ref

- An End-to-End Trainable Neural Network for Image-based Sequence Recognition
- CRNN (Tom)
- Handwriting Recognition with ML
- CTC Loss (Alex Graves)

THANKS FOR LISTENING