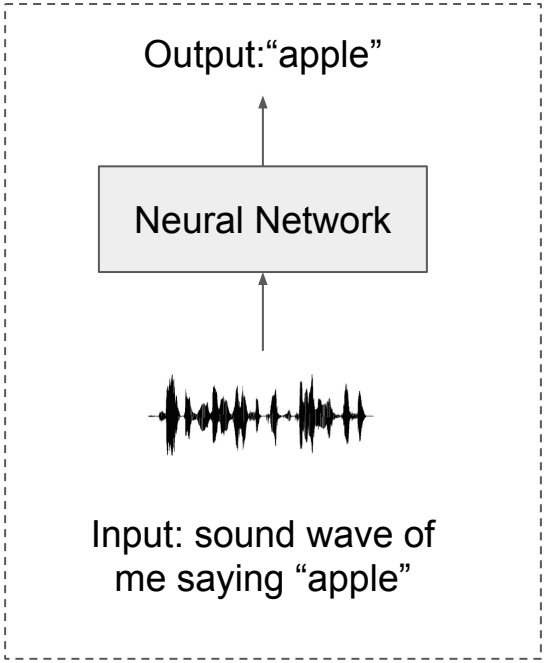
Review from deepsystems.ai

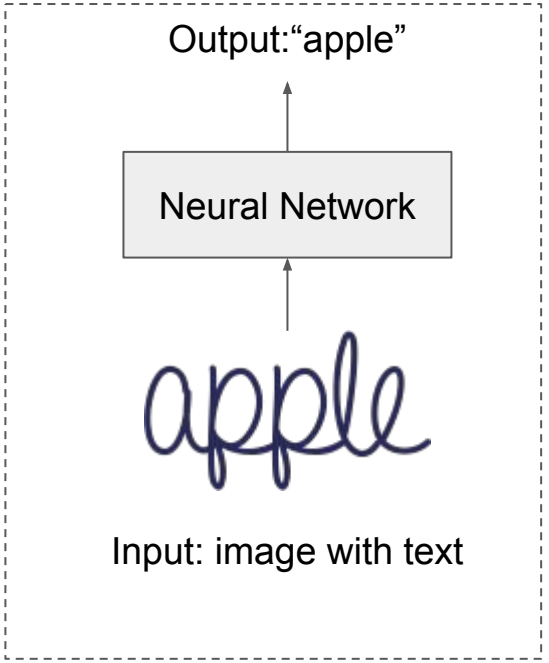
How to build end-to-end recognition system: best practices

Task Description

Labelling unsegmented sequence data. i.e. training data is not pre-segmented.



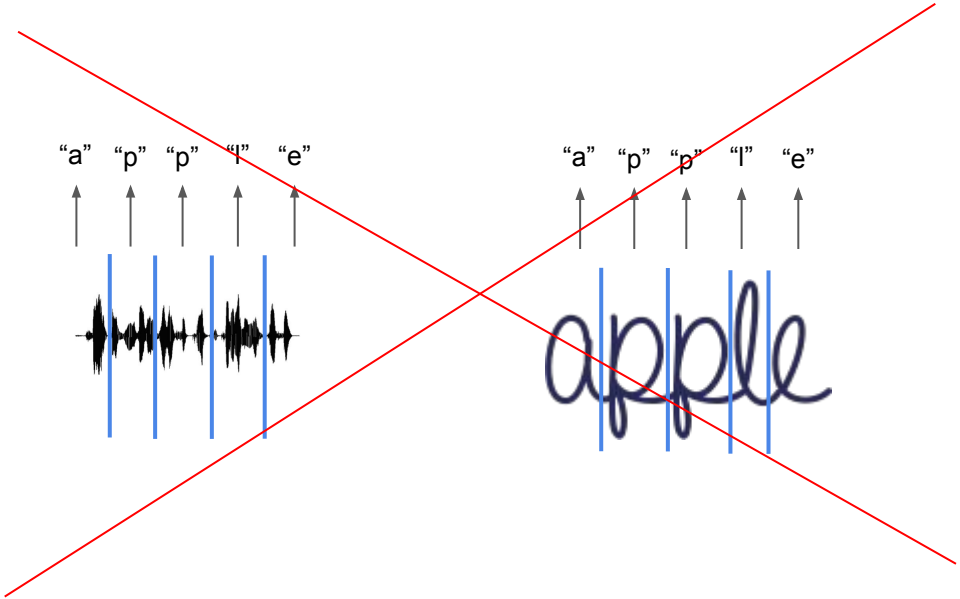
Example: speech recognition



Example: image ocr

Task Description

Labelling unsegmented sequence data. i.e. training data is not pre-segmented.



We can not pre-segment input data because:

- It is too time consuming
- It is too expensive
- It is impossible in most cases

Links

Alex Graves. CTC Loss: http://www.cs.toronto.edu/~graves/icml_2006.pdf

Keras example: image_ocr.ipynb

Big picture

Google Voice Search
Gets Smarter

Dropbox

[Google: voice search]

[Baidu: Deep Speech]

[Dropbox: document scanner]

Image OCR: model architecture

High-level overview

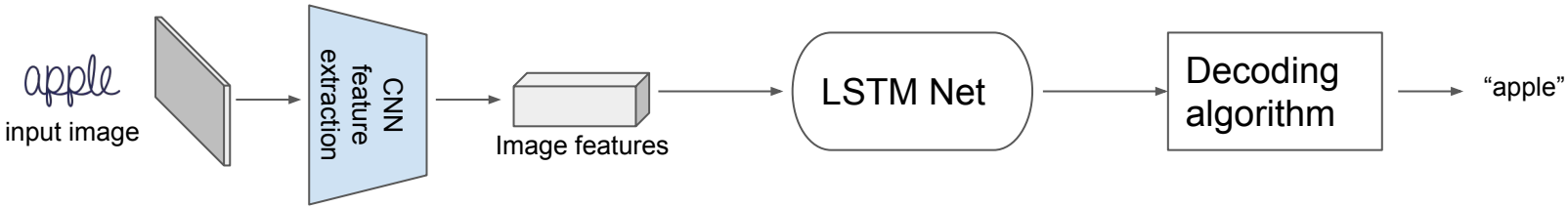
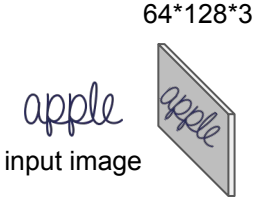
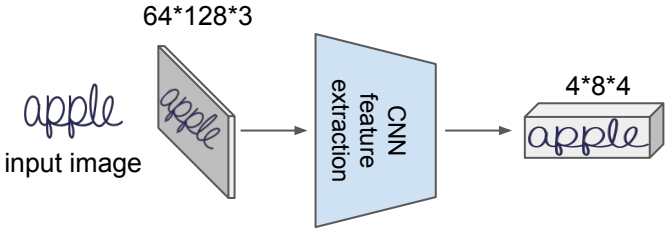
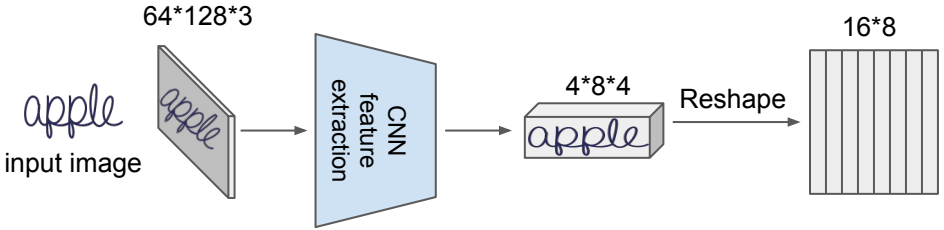

apple
input image
CNN feature extraction
Image features
LSTM Net
Decoding algorithm
“apple”

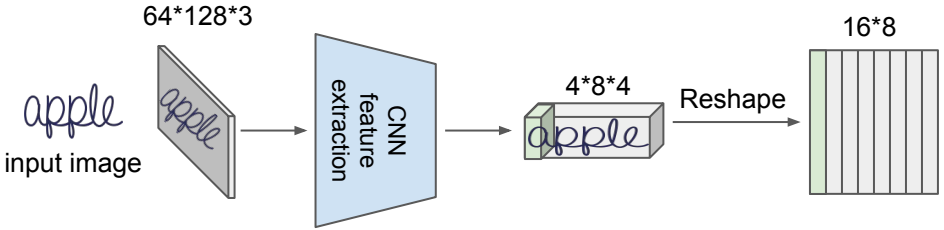
Image OCR: model architecture

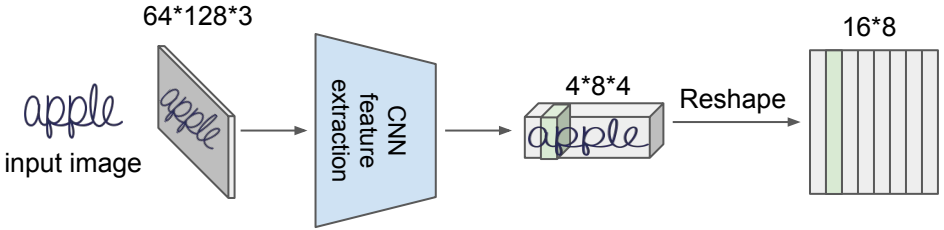
Detailed overview

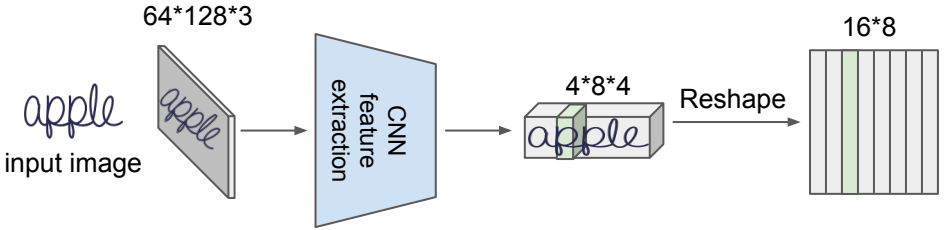

64*128*3
apple
input image
apple

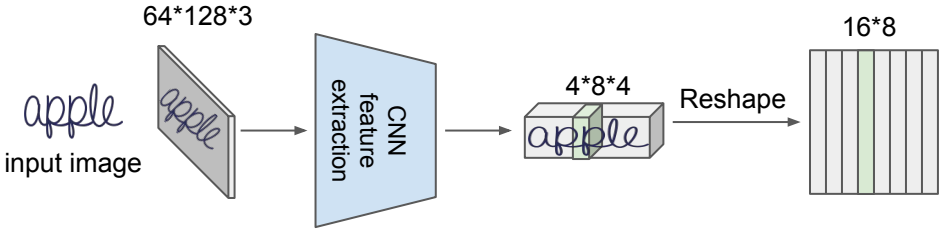

64*128*3
apple
input image
apple
CNN
feature
extraction
4*8*4
apple

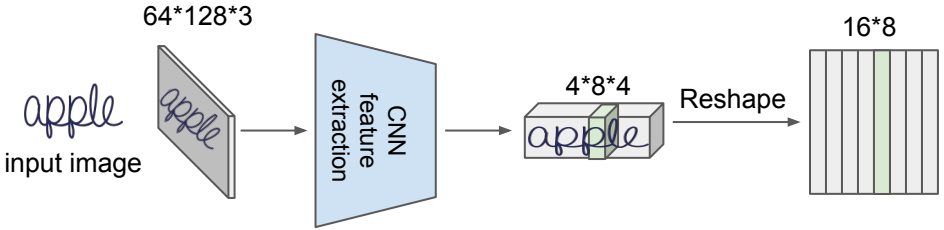

64*128*3
apple
input image
CNN
feature
extraction
4*8*4
apple
Reshape
16*8

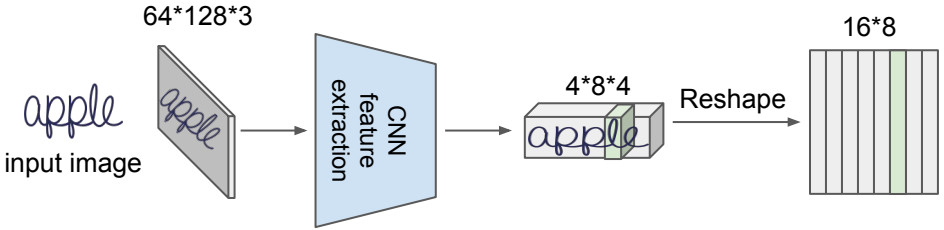

64*128*3
apple
input image
CNN
feature
extraction
4*8*4
Reshape
16*8

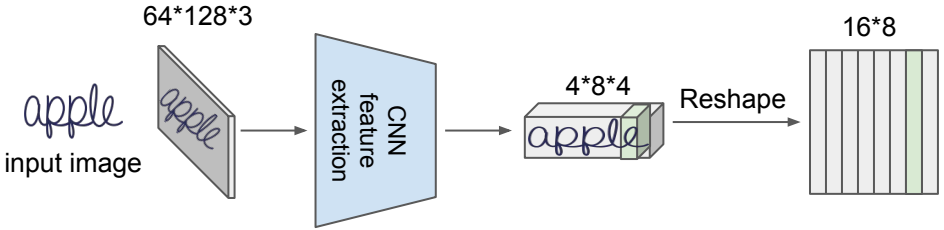

64*128*3
apple
input image
CNN feature extraction
4*8*4
Reshape
16*8

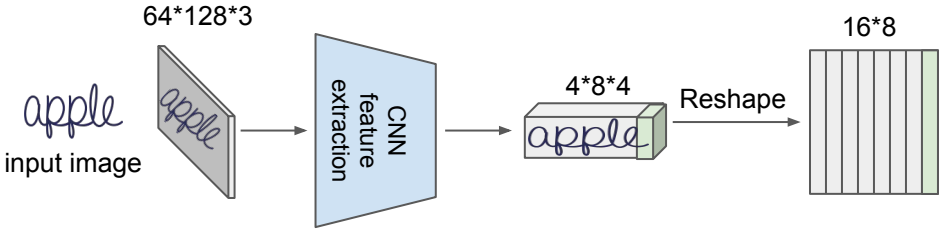

64*128*3
input image
CNN feature extraction
4*8*4
Reshape
16*8

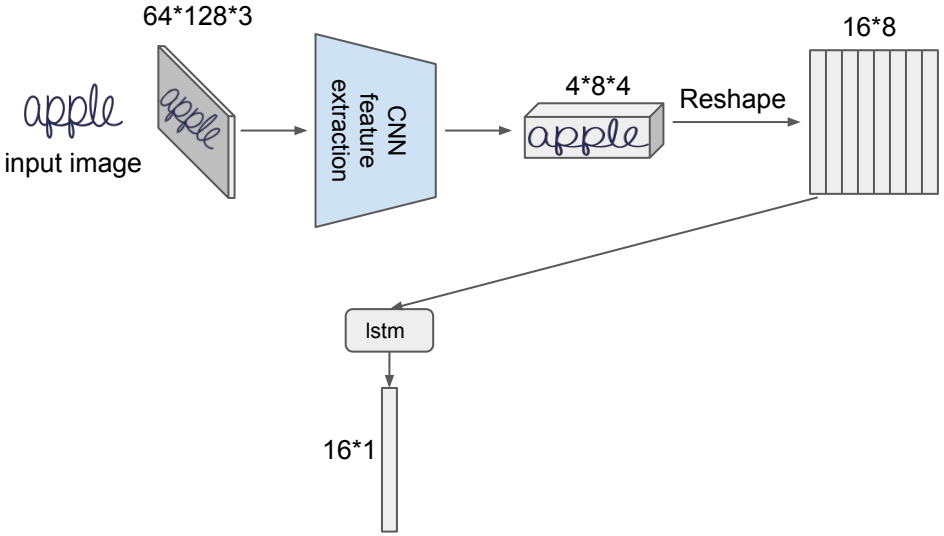

64*128*3
apple
input image
CNN feature extraction
4*8*4
Reshape
16*8

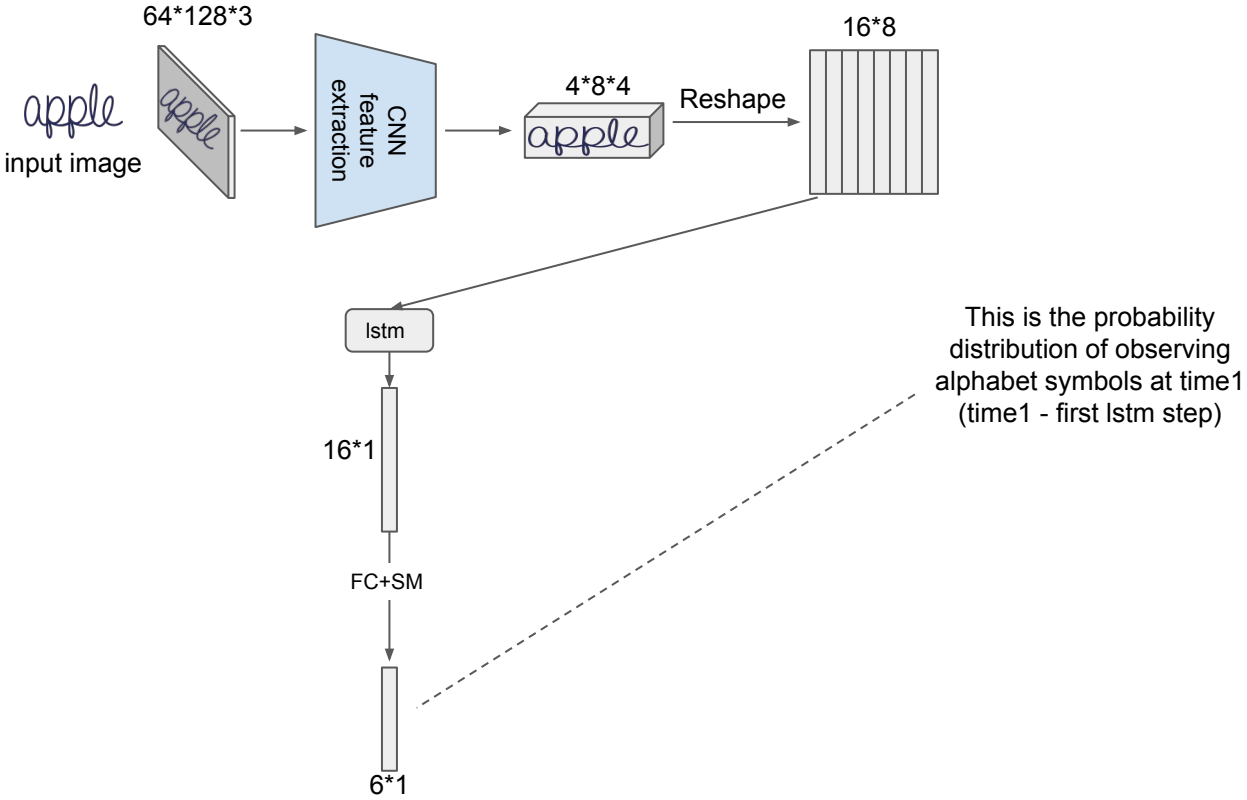

64*128*3
apple
input image
CNN feature extraction
4*8*4
Reshape
16*8

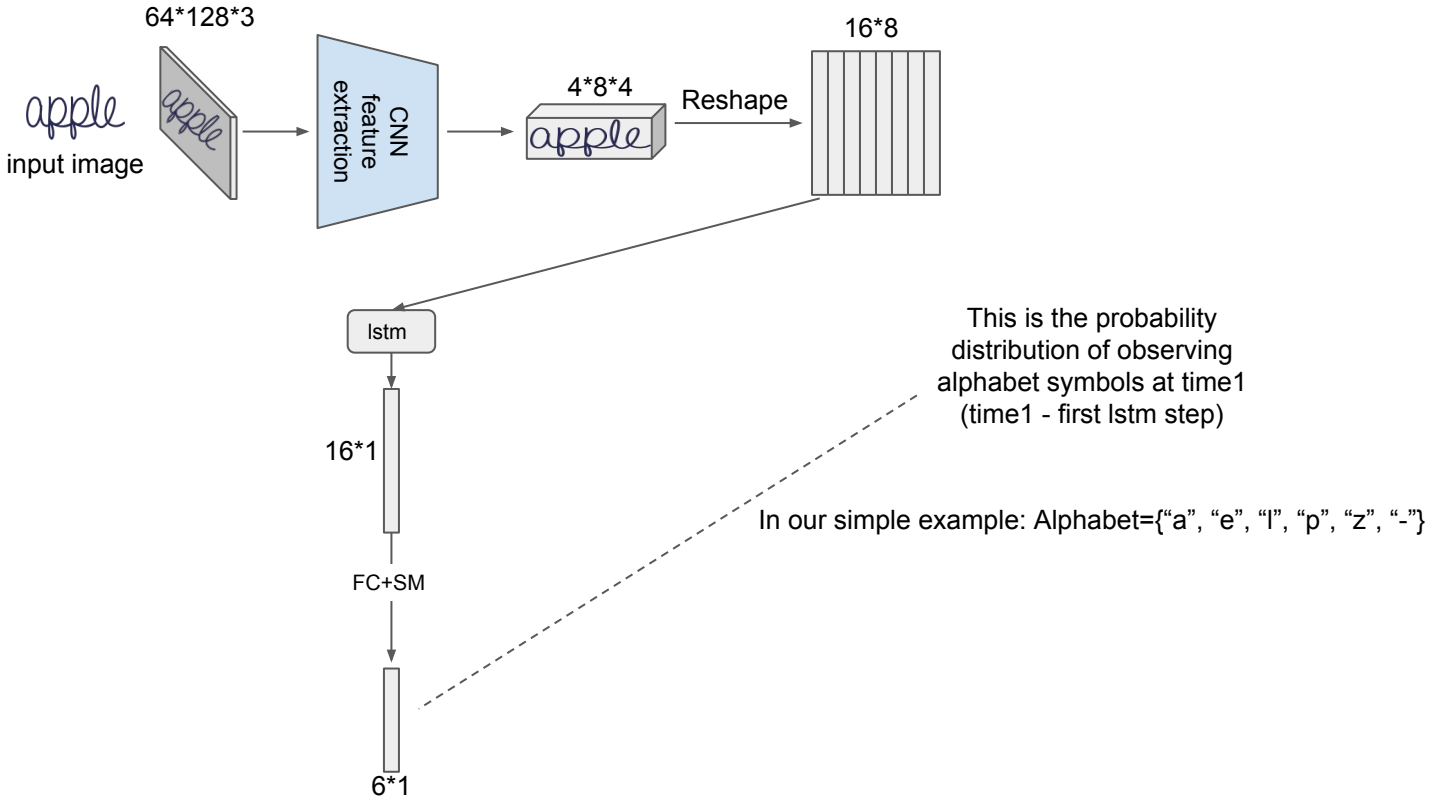

64*128*3
apple
input image
CNN feature extraction
4*8*4
Reshape
16*8

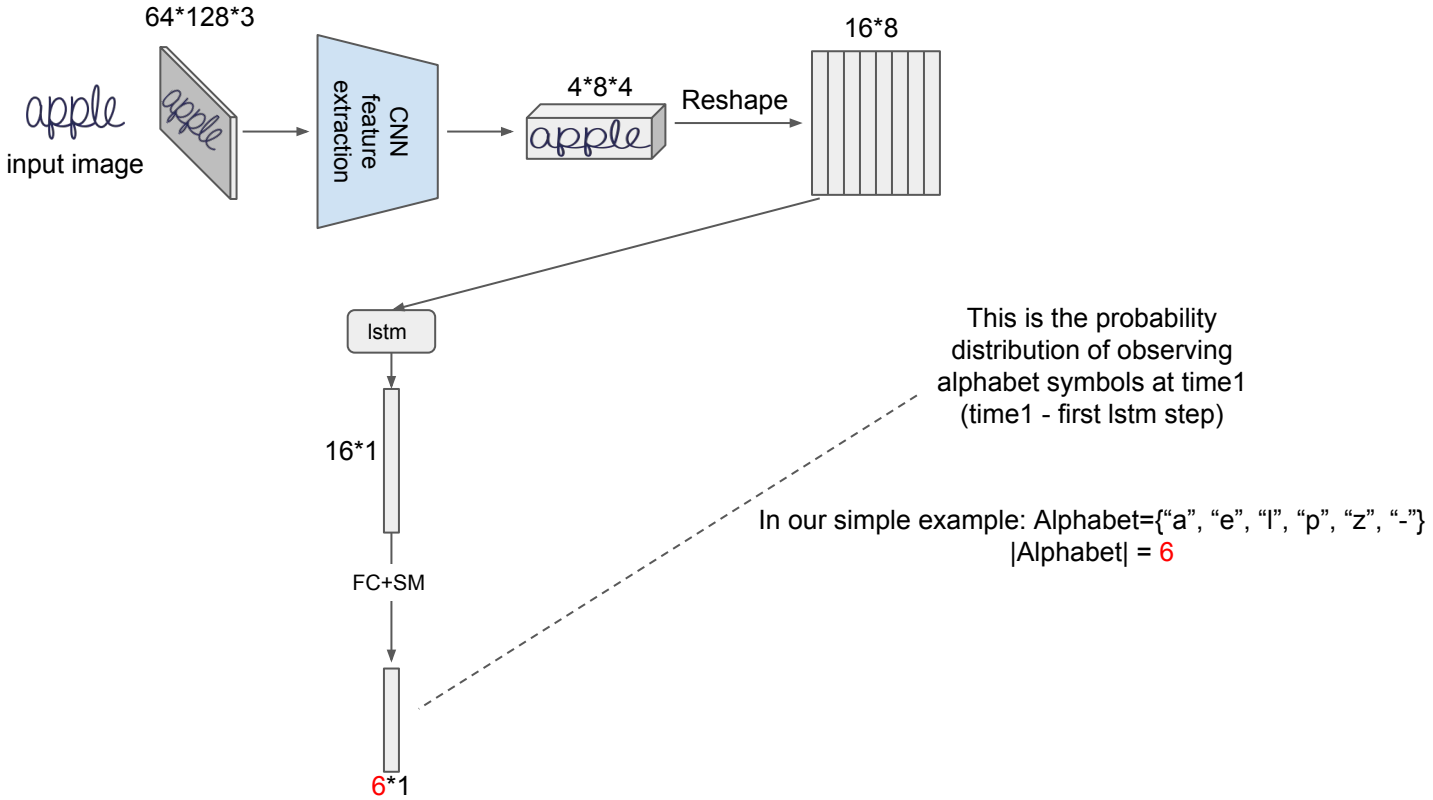

64*128*3
apple
input image
apple
CNN
feature
extraction
4*8*4
apple
Reshape
16*8

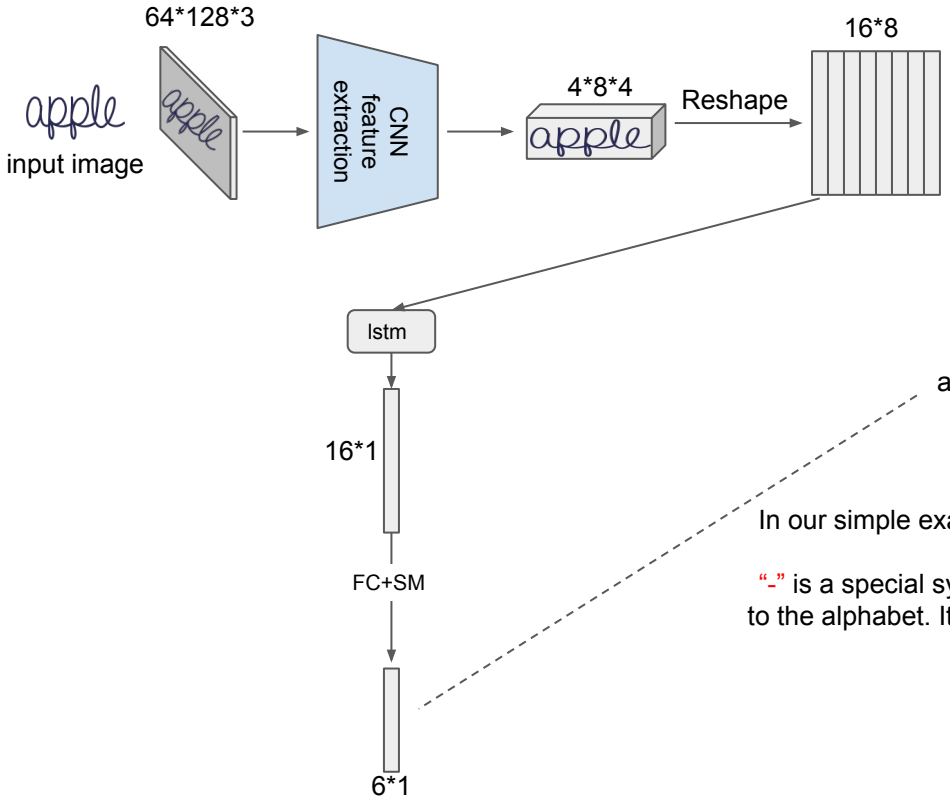

64*128*3
apple
input image
CNN feature extraction
4*8*4
Reshape
16*8


64*128*3
input image
CNN feature extraction
4*8*4
Reshape
16*8
lstm
16*1


64*128*3
input image
CNN feature extraction
4*8*4
Reshape
16*8
lstm
16*1
FC+SM
6*1
This is the probability distribution of observing alphabet symbols at time1 (time1 - first lstm step)

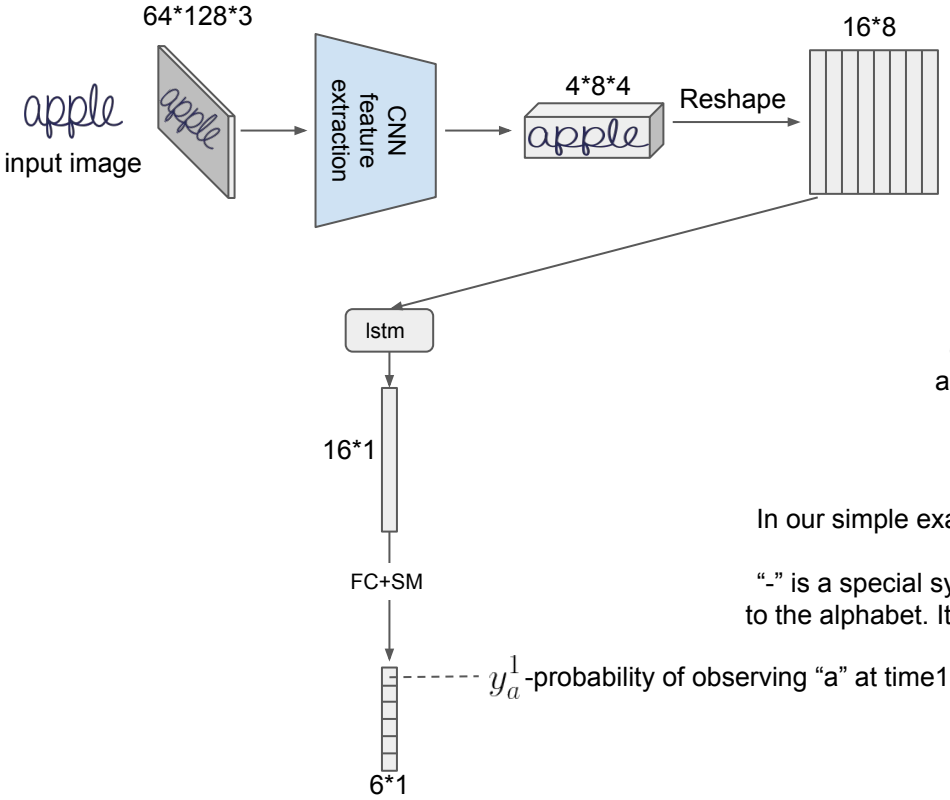

64*128*3
input image
CNN feature extraction
4*8*4
Reshape
16*8
lstm
16*1
FC+SM
6*1
This is the probability distribution of observing alphabet symbols at time1 (time1 - first lstm step)
In our simple example: Alphabet={“a”, “e”, “l”, “p”, “z”, “-”}


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



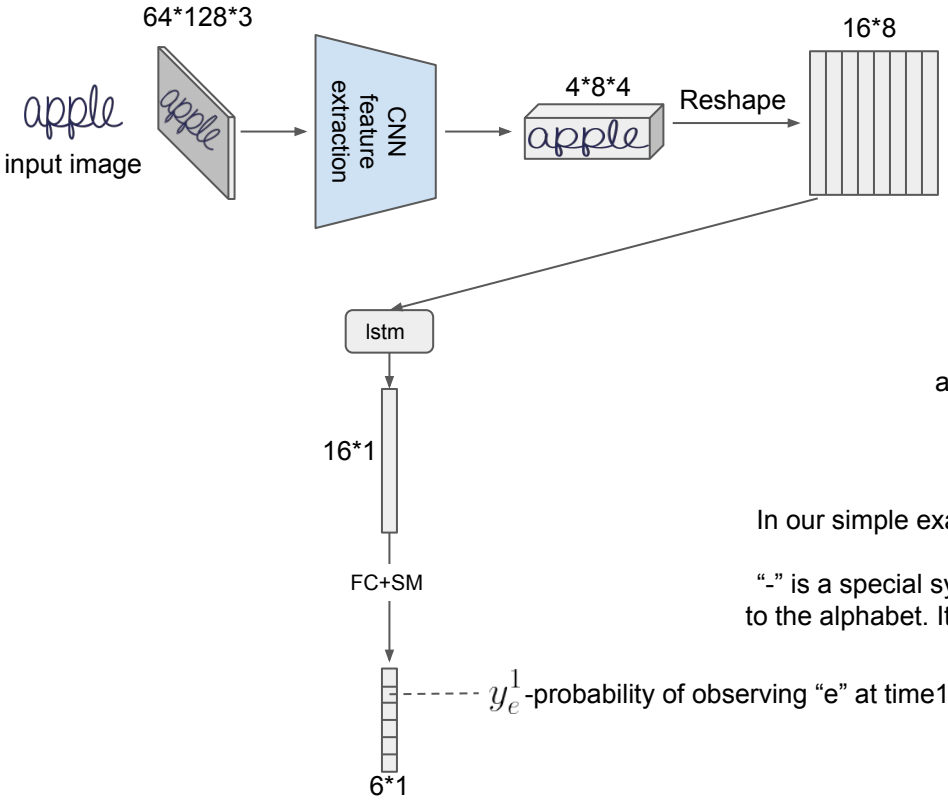
This is the probability distribution of observing alphabet symbols at time1 (time1 - first lstm step)

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.



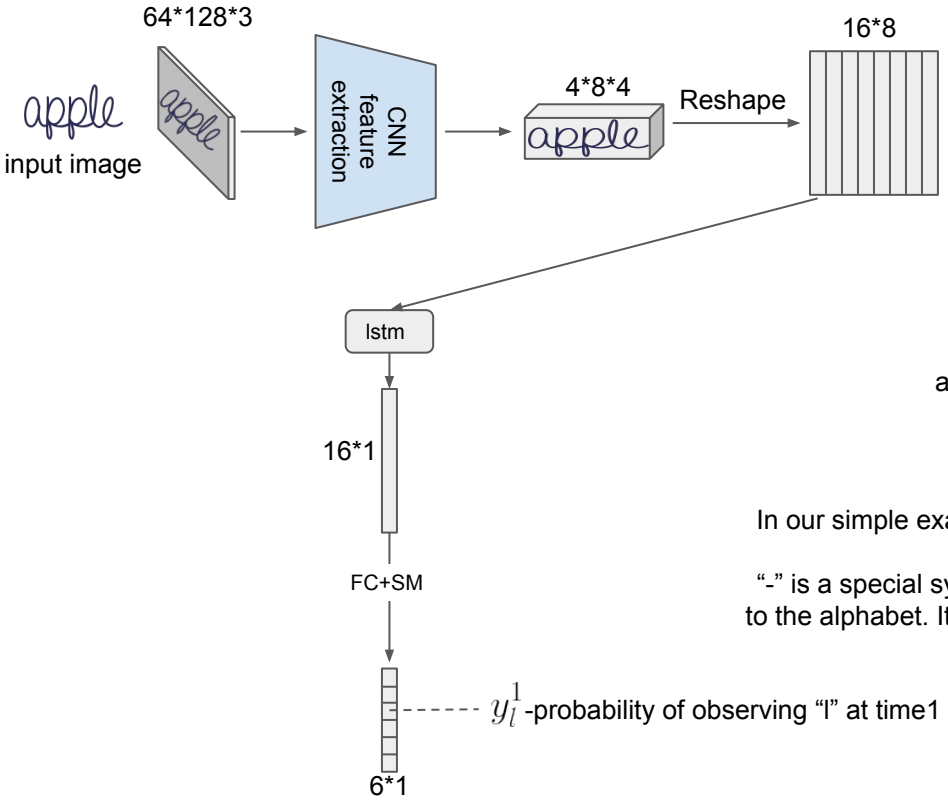
This is the probability distribution of observing alphabet symbols at time1 (time1 - first lstm step)

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.



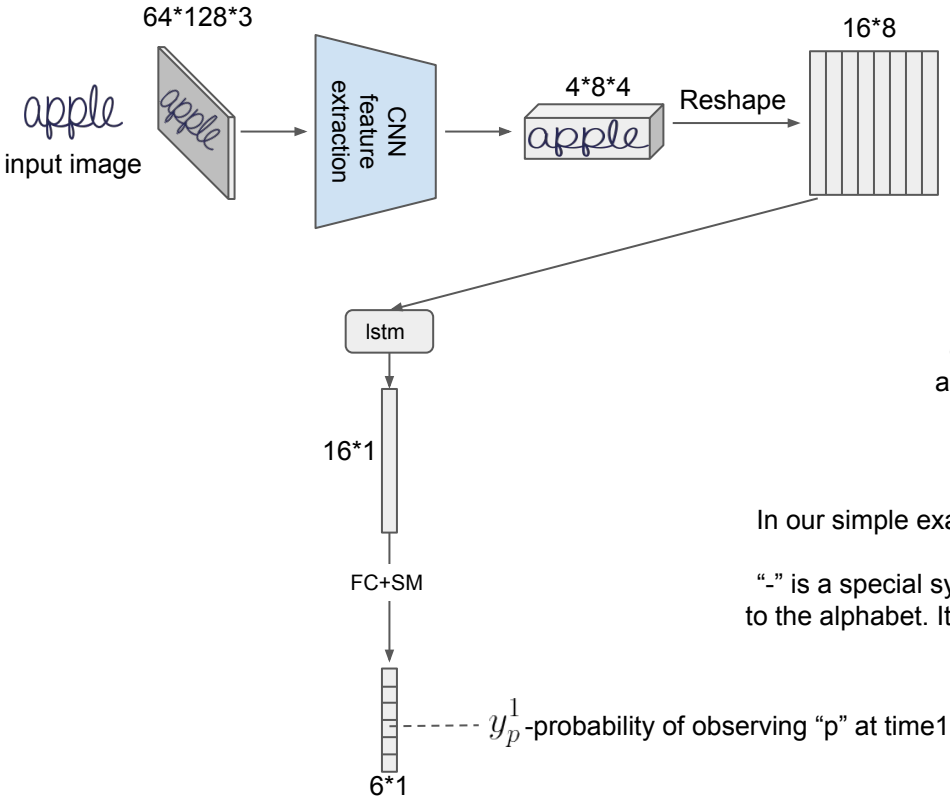
This is the probability distribution of observing alphabet symbols at time1 (time1 - first lstm step)

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.



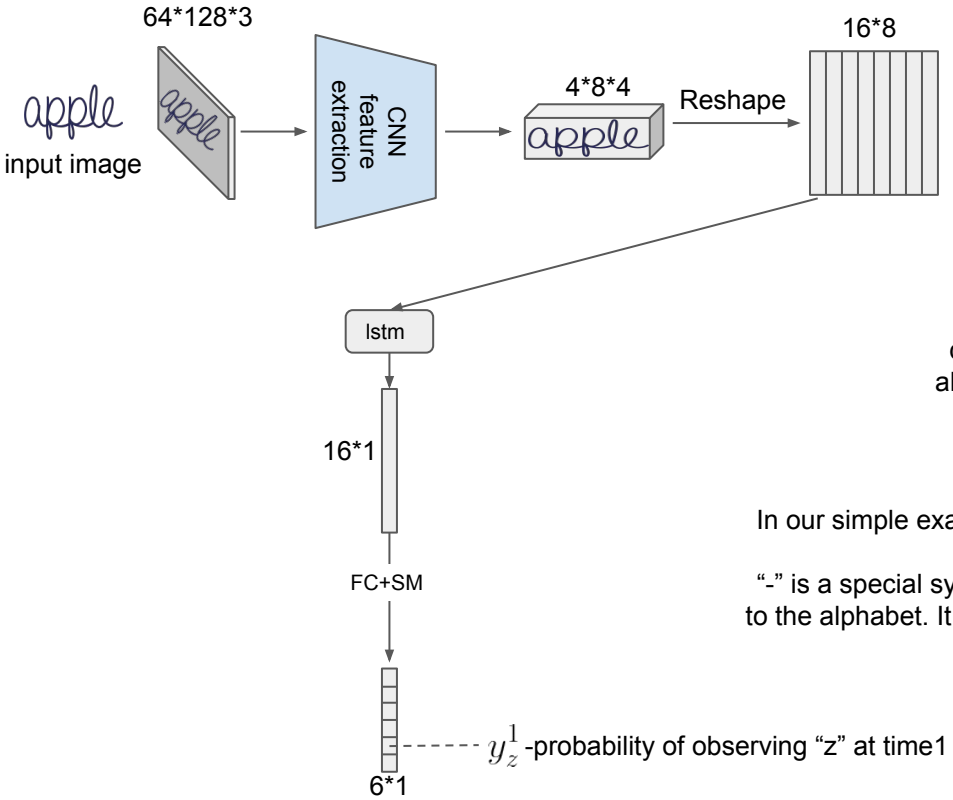
This is the probability distribution of observing alphabet symbols at time1 (time1 - first lstm step)

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.



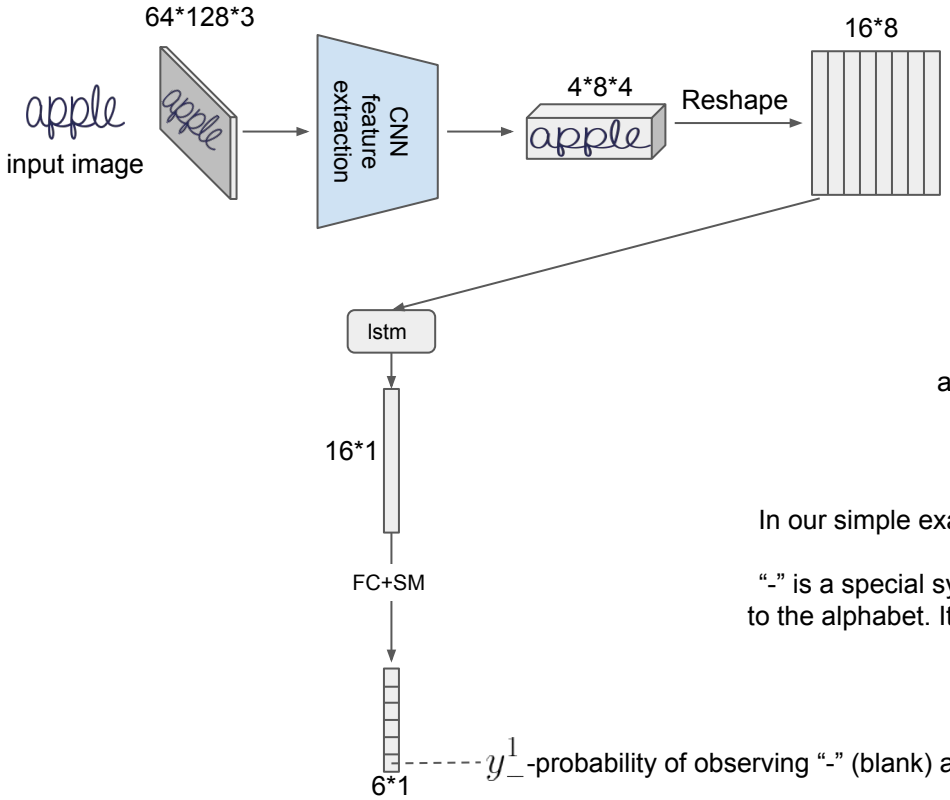
This is the probability distribution of observing alphabet symbols at time1 (time1 - first lstm step)

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.



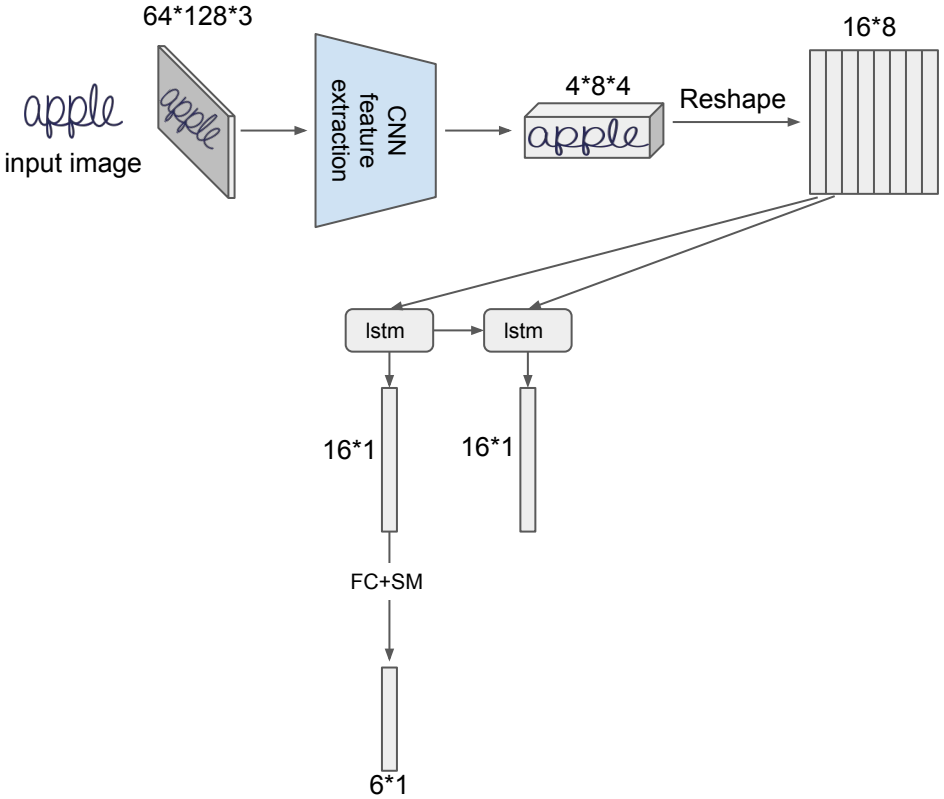
This is the probability distribution of observing alphabet symbols at time1 (time1 - first lstm step)

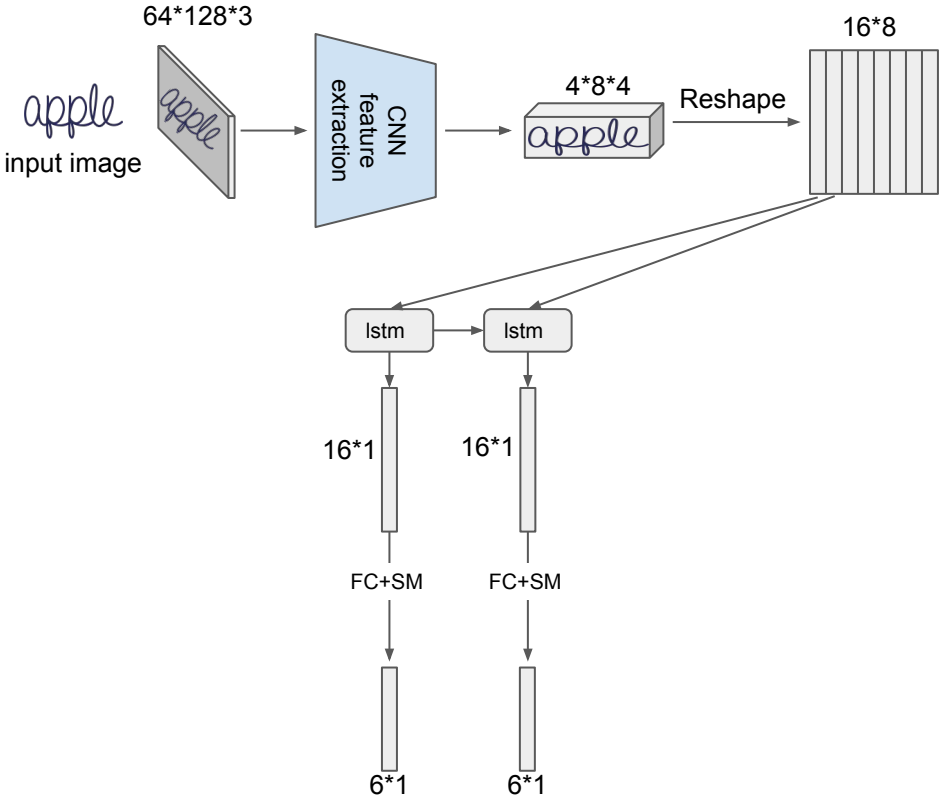
In our simple example: Alphabet={“a”, “e”, “l”, “p”, “z”, “-”}
|Alphabet| = 6
“-” is a special symbol (blank) that we always should add to the alphabet. It will be further understood what it is used for.

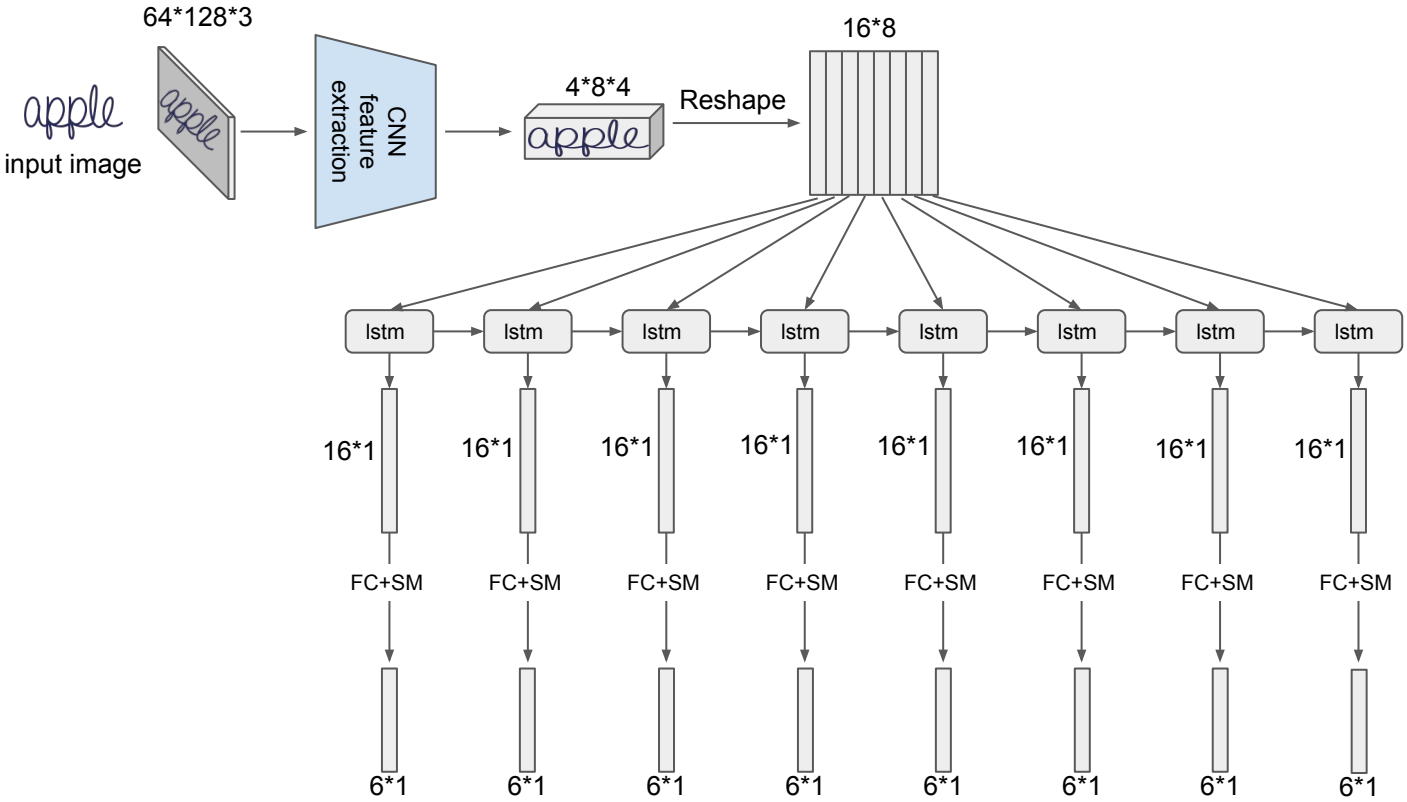


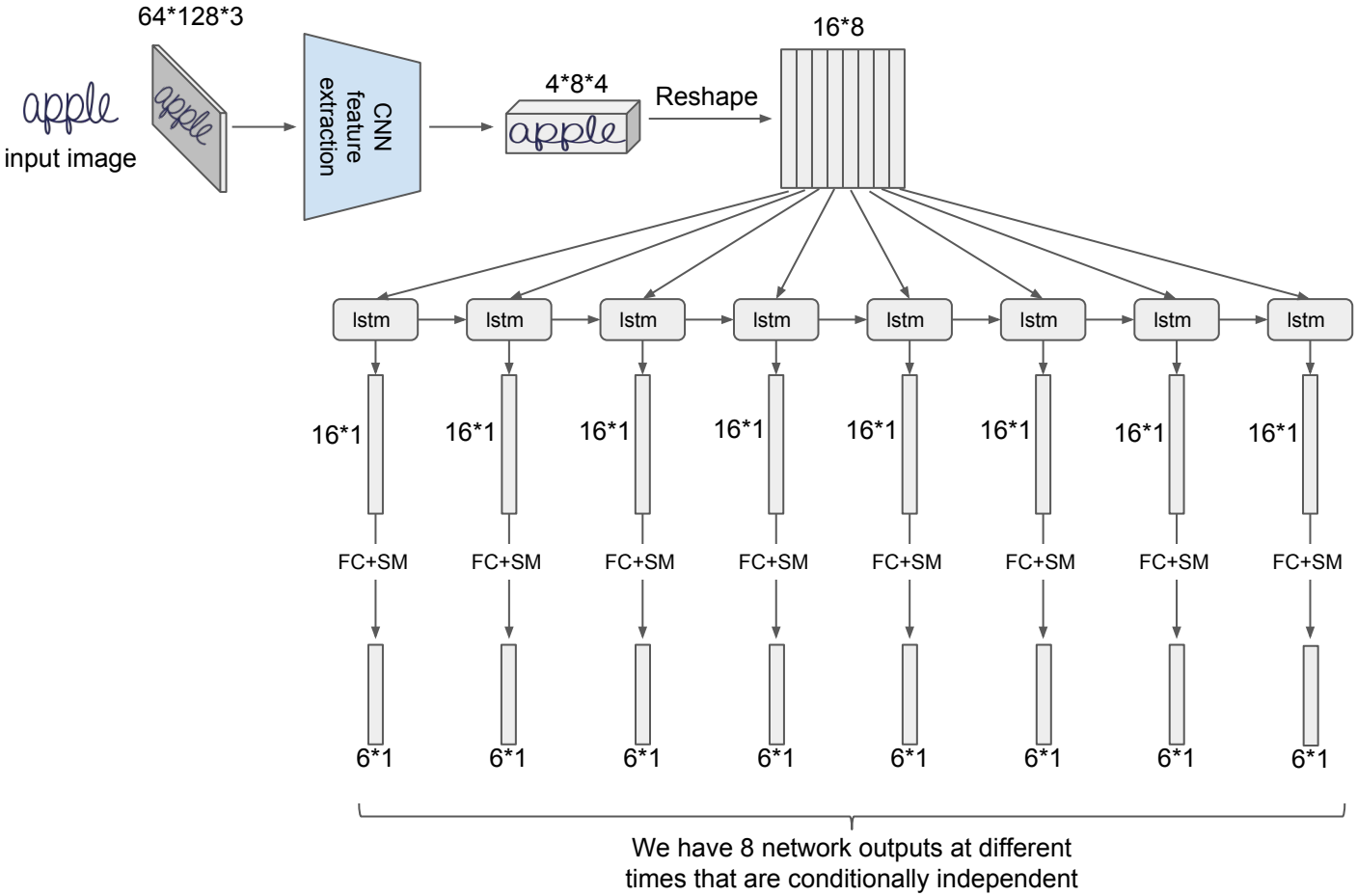
This is the probability distribution of observing alphabet symbols at time1 (time1 - first lstm step)

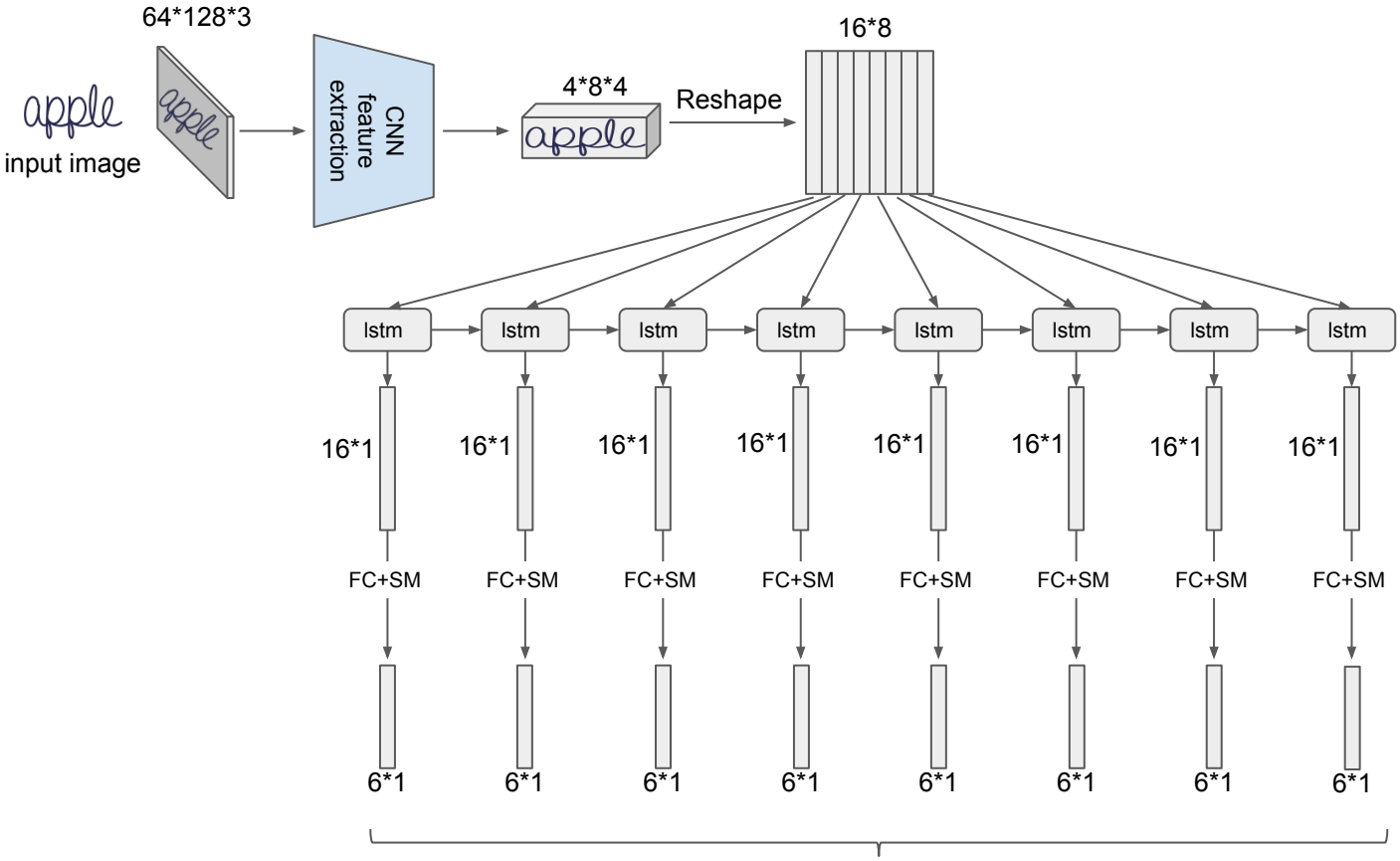
In our simple example: Alphabet={"a", "e", "l", "p", "z", "-"}
|Alphabet| = 6
"-" is a special symbol (blank) that we always should add to the alphabet. It will be further understood what it is used for.


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

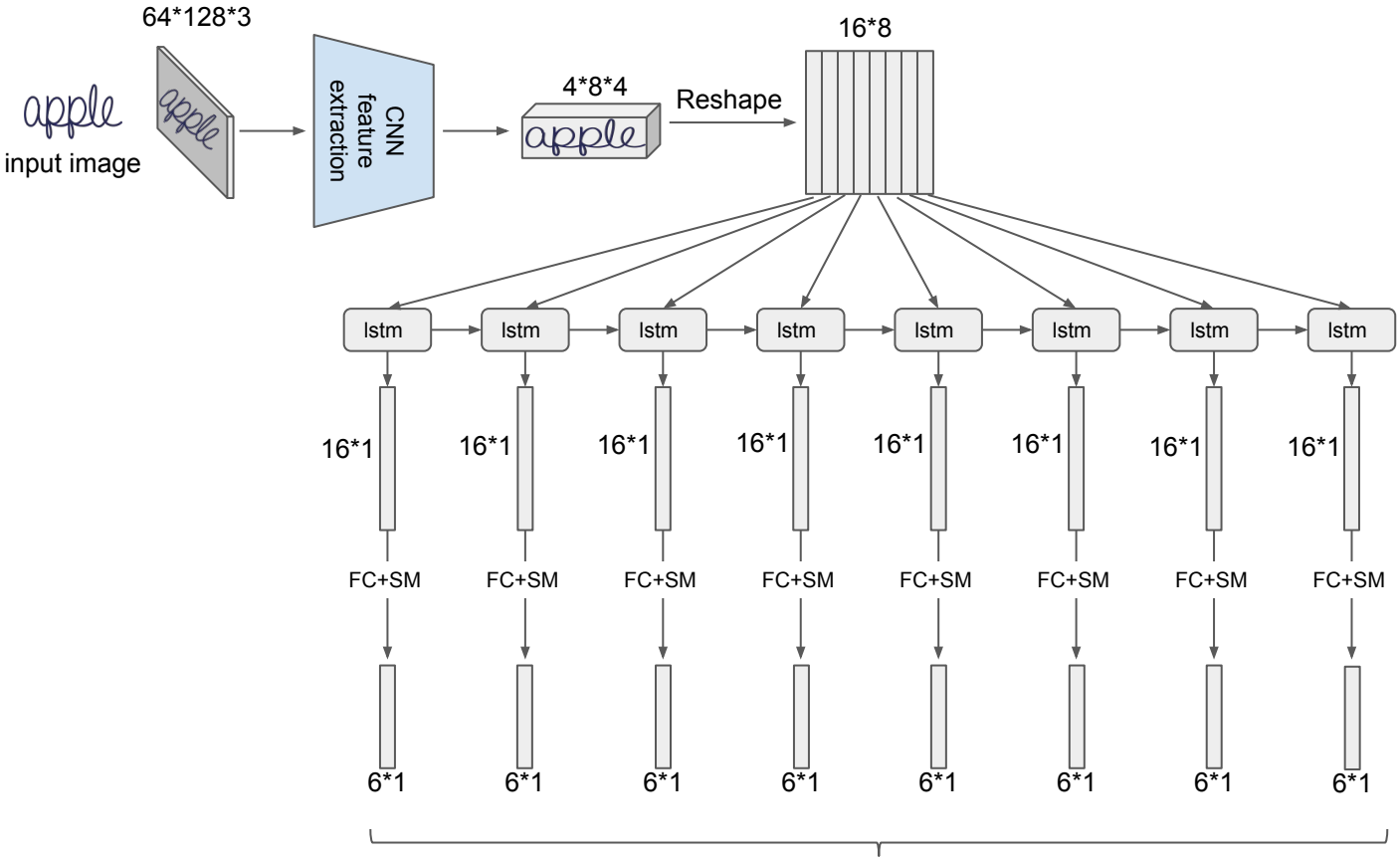

64*128*3
apple
input image
CNN feature extraction
4*8*4
Reshape
16*8
lstm
lstm
16*1
16*1
FC+SM
FC+SM
6*1
6*1


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


64*128*3
input image
CNN feature extraction
4*8*4
Reshape
16*8
lstm
16*1
FC+SM
6*1
We have 8 network outputs at different times that are conditionally independent

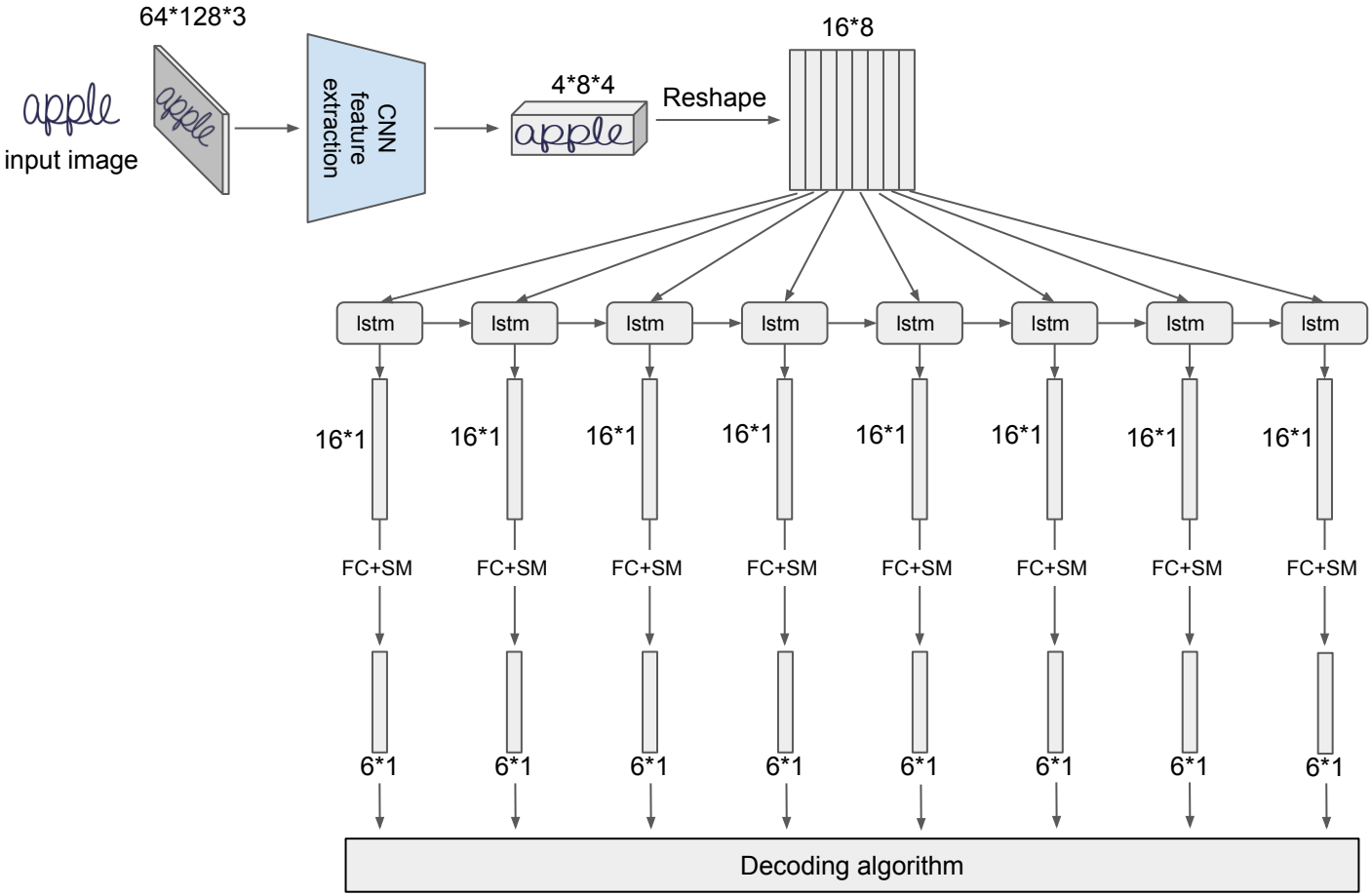


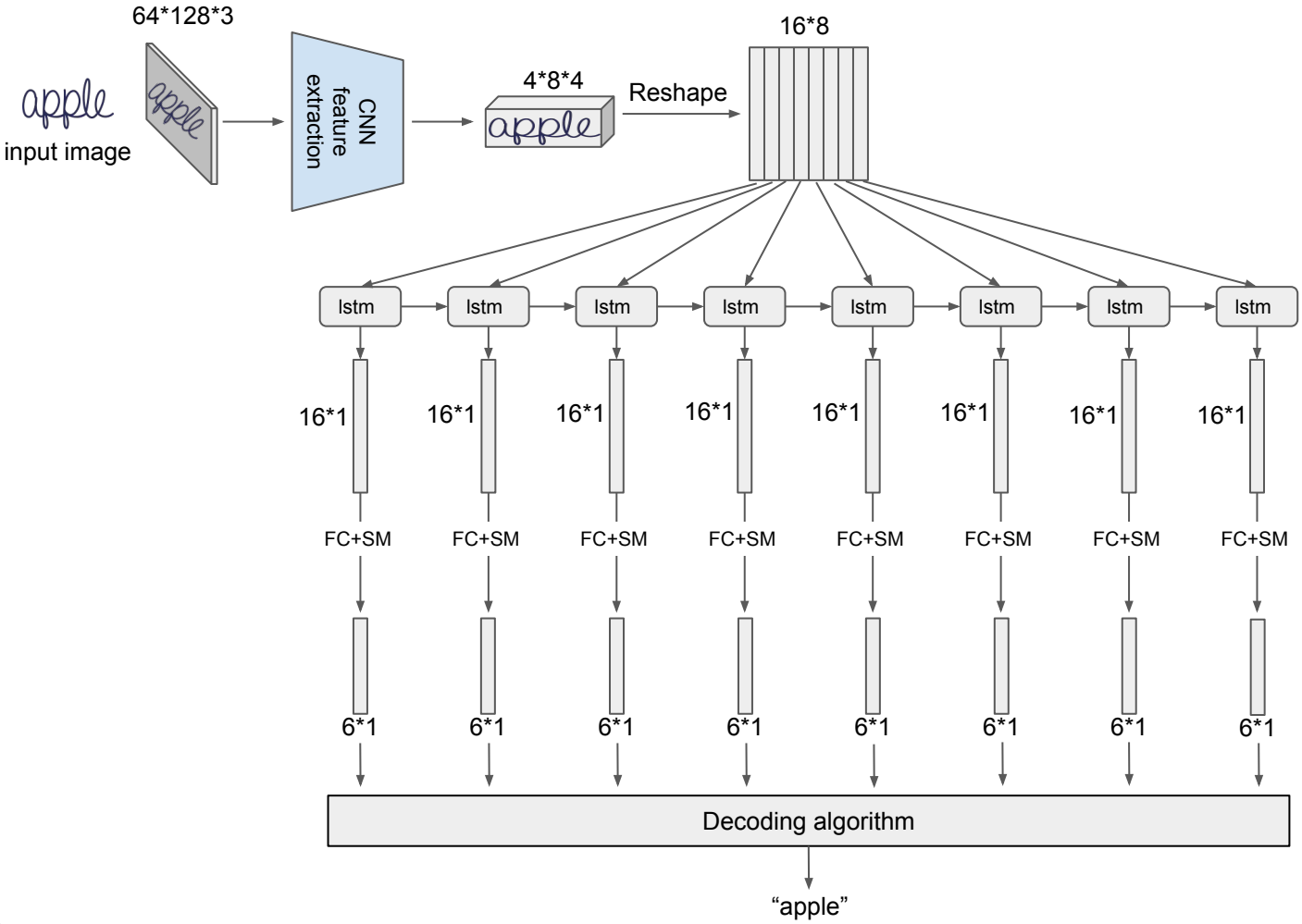
Note: We designed simplified neural network to have 8 outputs. It means that we can not recognize more than 8 characters per image.

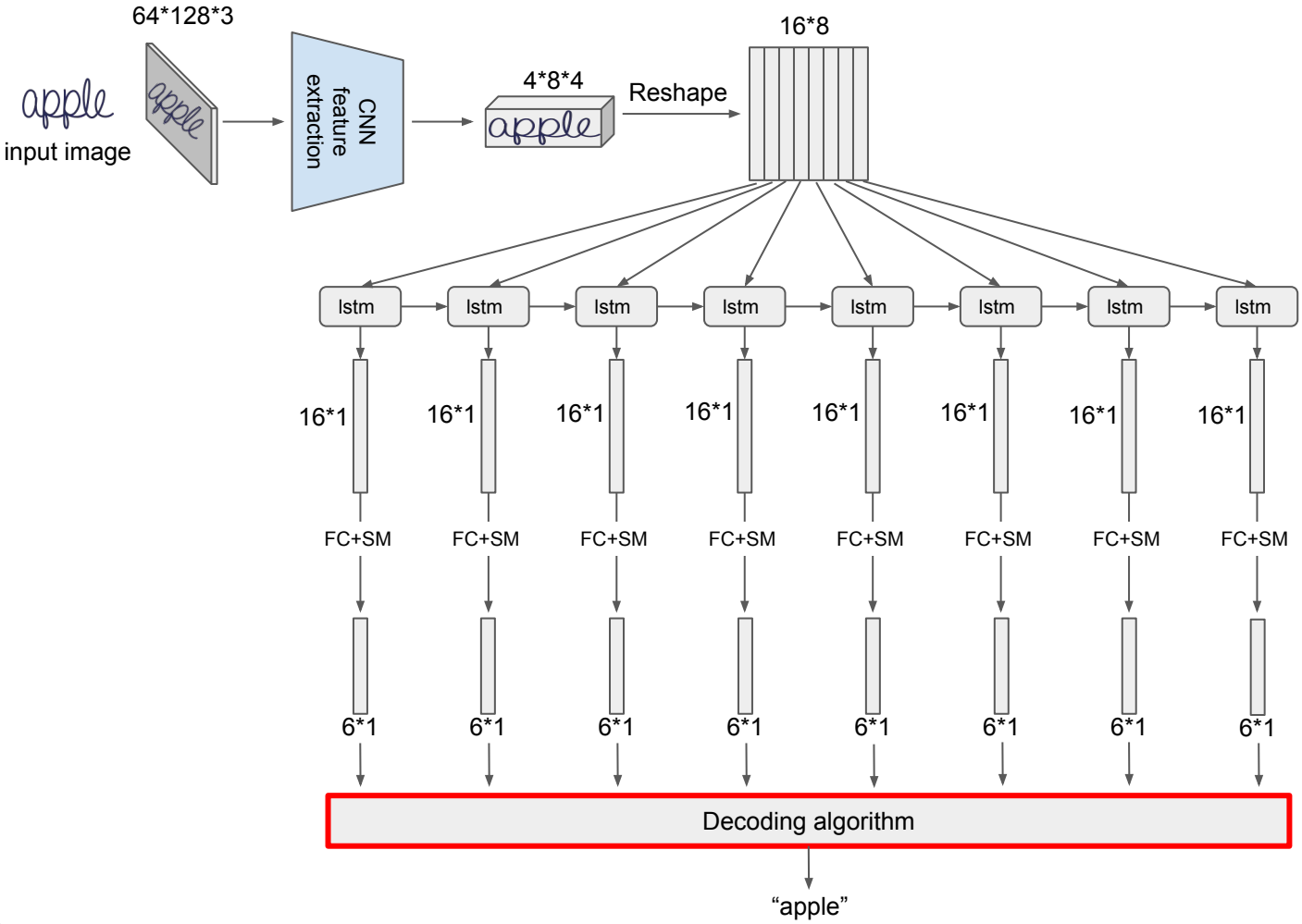


Note: We designed simplified neural network to have 8 outputs. It means that we can not recognize more than 8 characters per image.

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


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

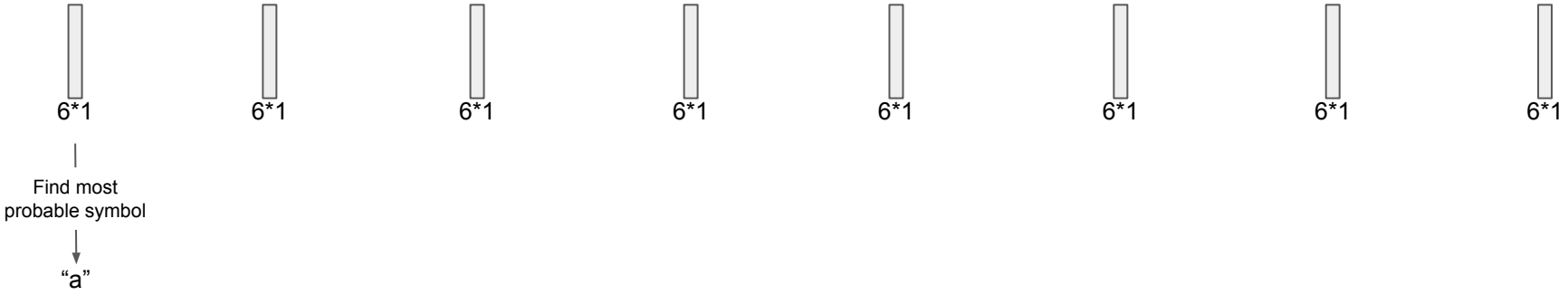


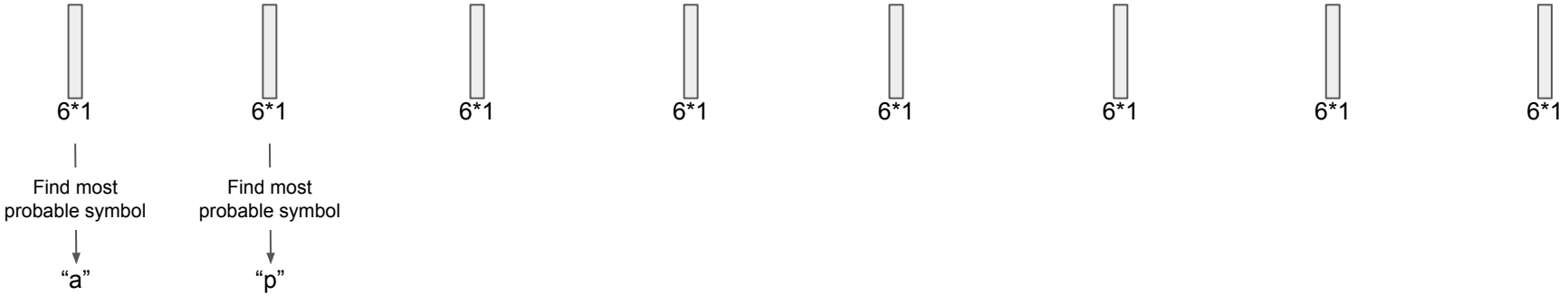
How does decoding algorithm work?

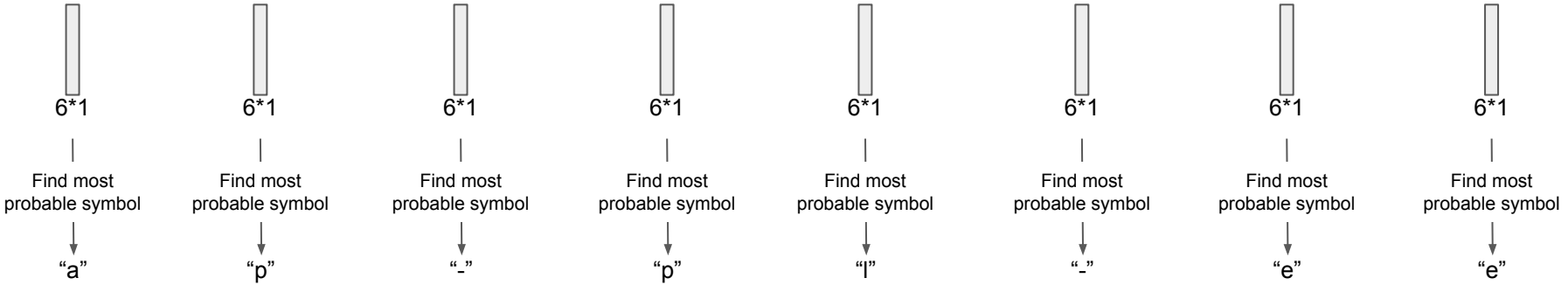
Image OCR: model architecture

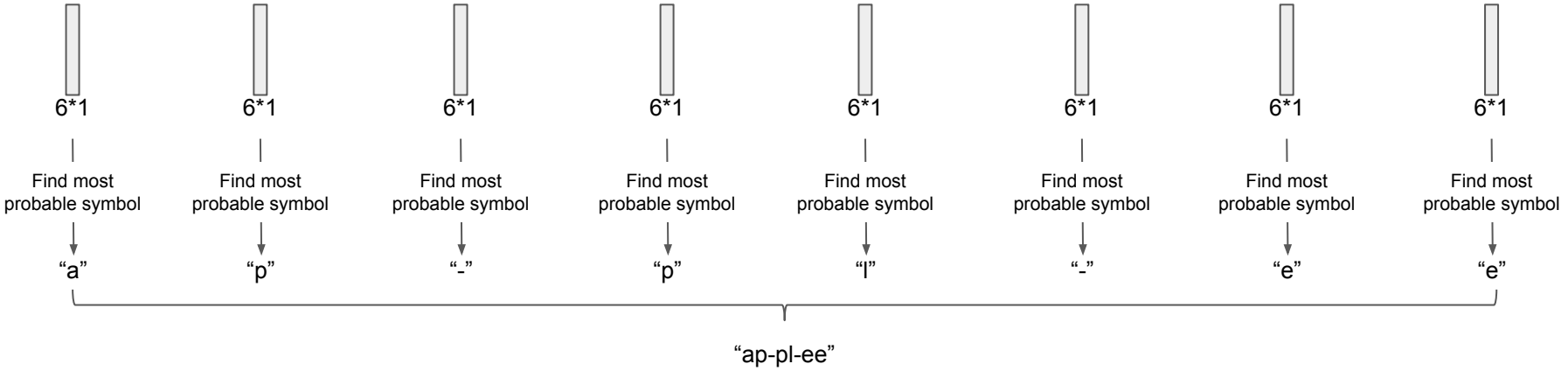
Decoding algorithm

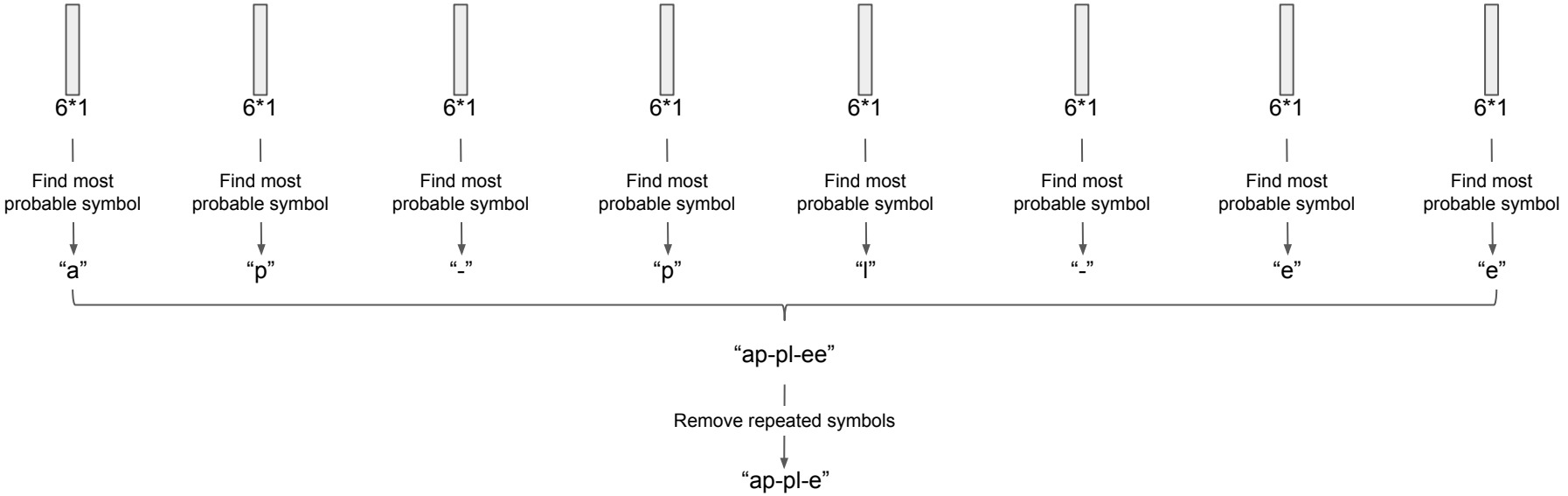
6*1
6*1
6*1
6*1
6*1
6*1
6*1
6*1

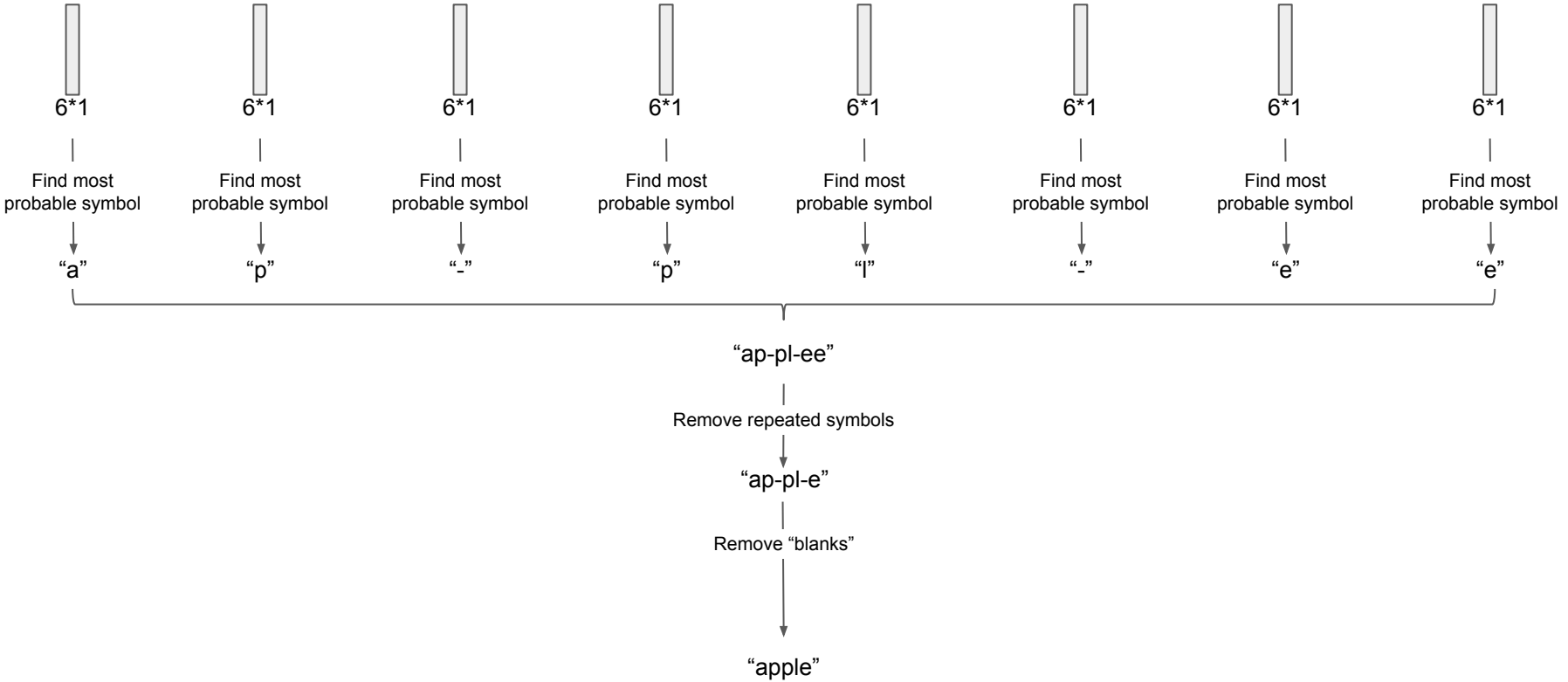

6*1
6*1
6*1
6*1
6*1
6*1
6*1
6*1
Find most probable symbol
“a”

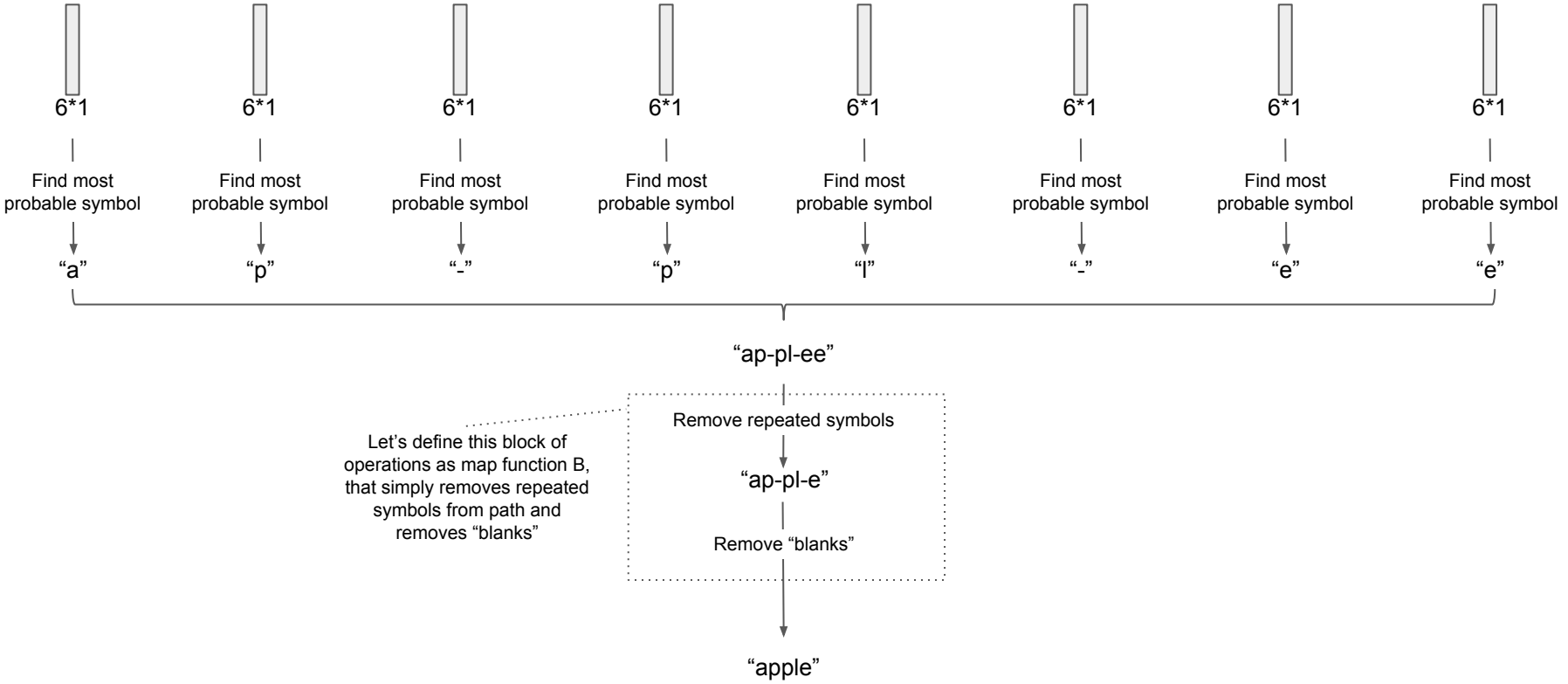

6*1
6*1
6*1
6*1
6*1
6*1
6*1
6*1
Find most probable symbol
Find most probable symbol
“a”
“p”

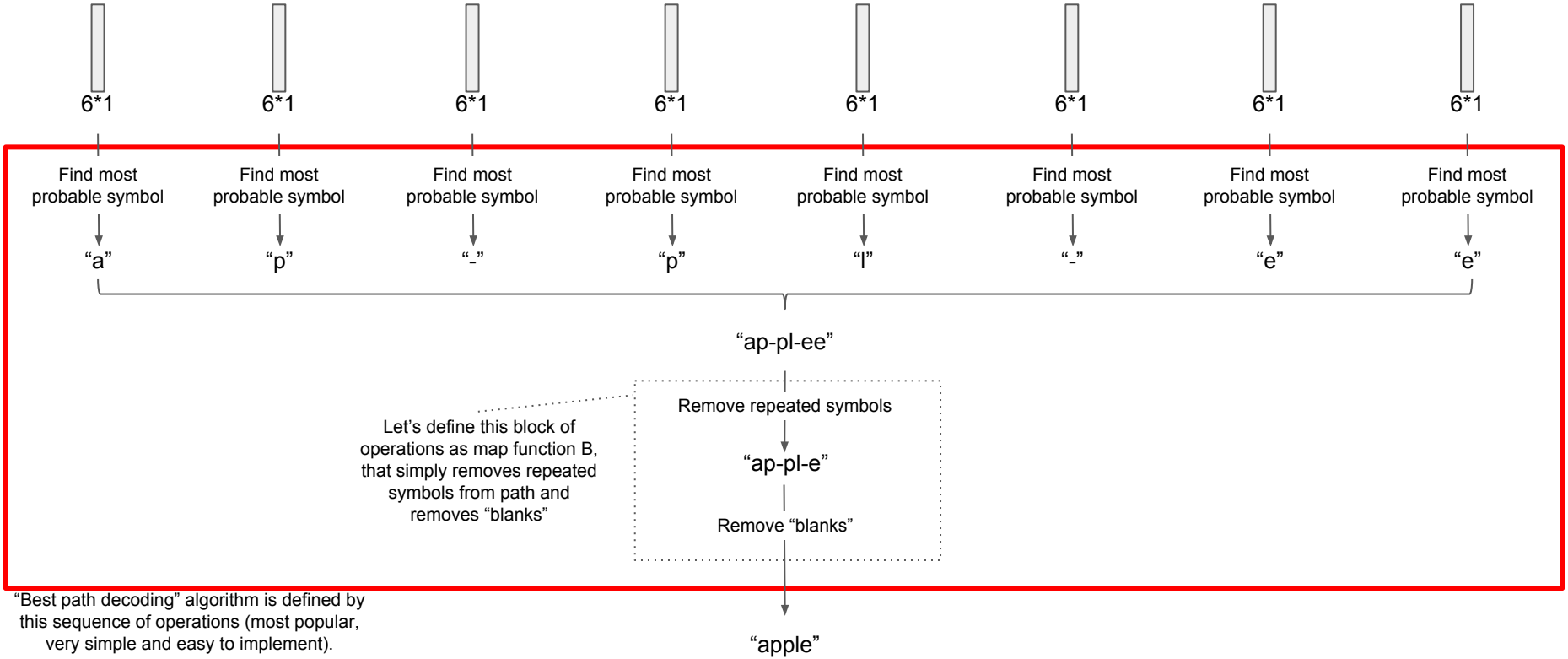

6*1
Find most probable symbol
“a”
6*1
Find most probable symbol
“p”
6*1
Find most probable symbol
“-”
6*1
Find most probable symbol
“p”
6*1
Find most probable symbol
“l”
6*1
Find most probable symbol
“-”
6*1
Find most probable symbol
“e”
6*1
Find most probable symbol
“e”


6*1
6*1
6*1
6*1
6*1
6*1
6*1
6*1
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
“a”
“p”
“-”
“p”
“l”
“-”
“e”
“e”
“ap-pl-ee”

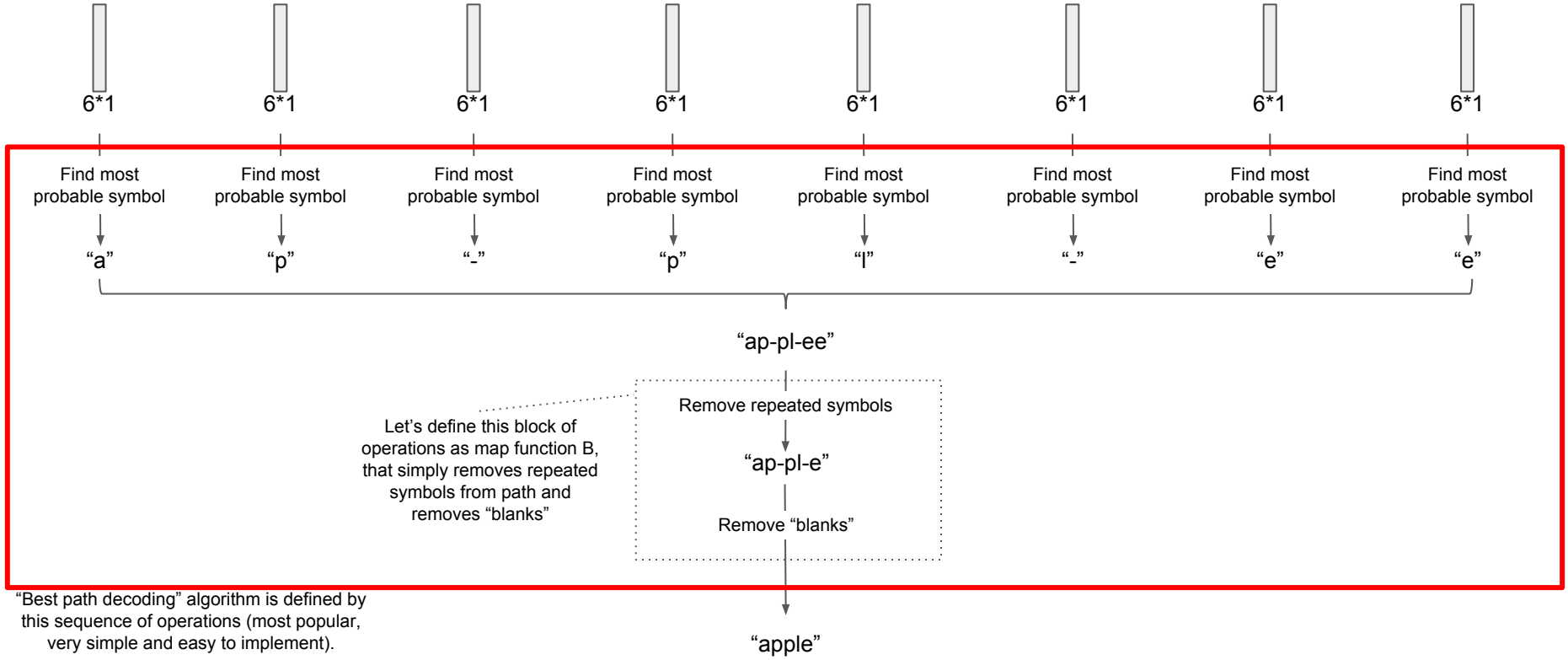

6*1
6*1
6*1
6*1
6*1
6*1
6*1
6*1
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
“a”
“p”
“-”
“p”
“l”
“-”
“e”
“e”
“ap-pl-ee”
Remove repeated symbols
“ap-pl-e”


6*1
6*1
6*1
6*1
6*1
6*1
6*1
6*1
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
“a”
“p”
“-”
“p”
“l”
“-”
“e”
“e”
“ap-pl-ee”
Remove repeated symbols
“ap-pl-e”
Remove “blanks”
“apple”


6*1
6*1
6*1
6*1
6*1
6*1
6*1
6*1
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
Find most probable symbol
“a”
“p”
“-”
“p”
“l”
“-”
“e”
“e”
“ap-pl-ee”
Let’s define this block of operations as map function B, that simply removes repeated symbols from path and removes “blanks”
Remove repeated symbols
“ap-pl-e”
Remove “blanks”
“apple”



“Best path decoding” algorithm is defined by this sequence of operations (most popular, very simple and easy to implement).



“Best path decoding” algorithm is defined by this sequence of operations (most popular, very simple and easy to implement).

Note: there are a few other algorithms in literature.

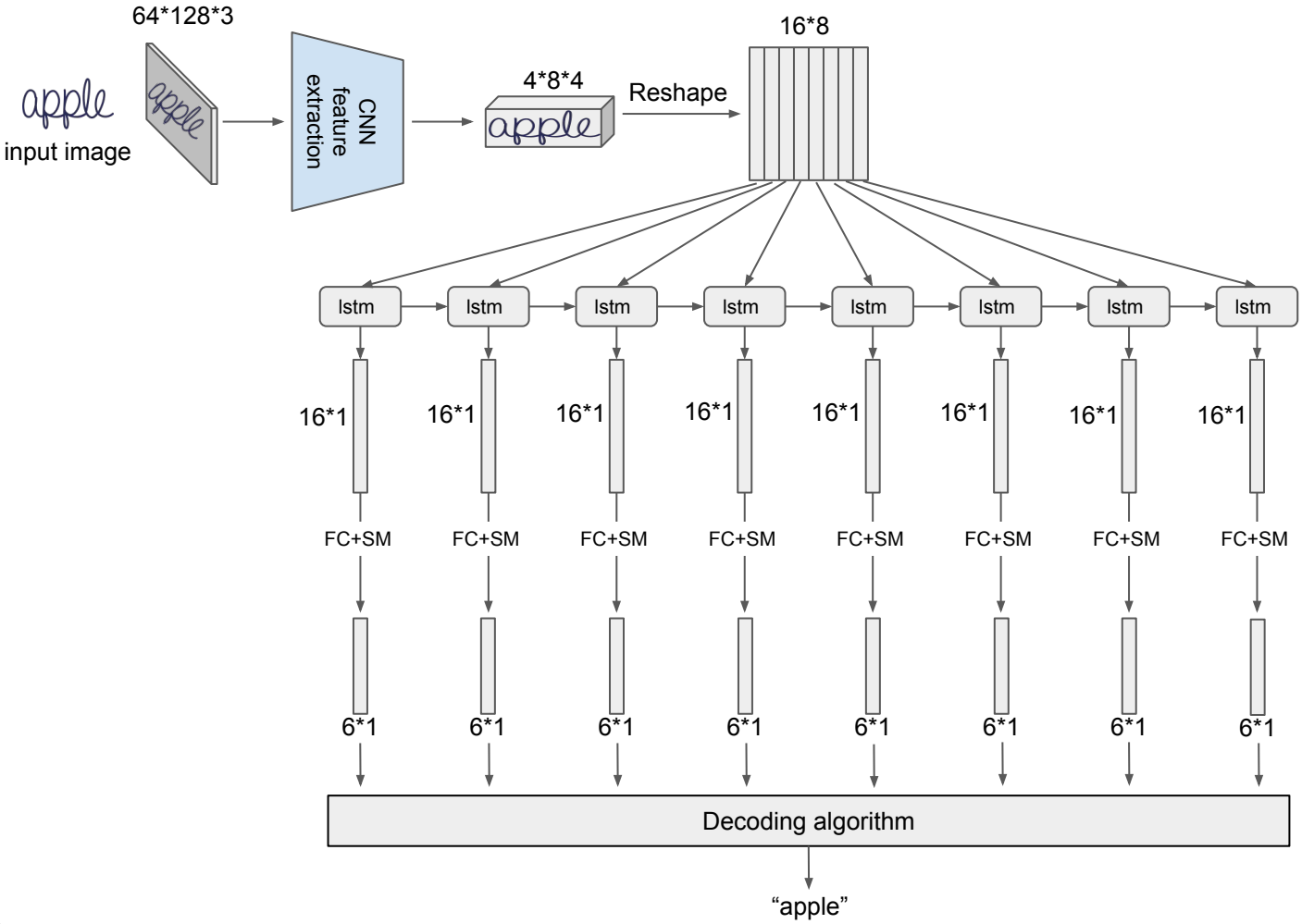
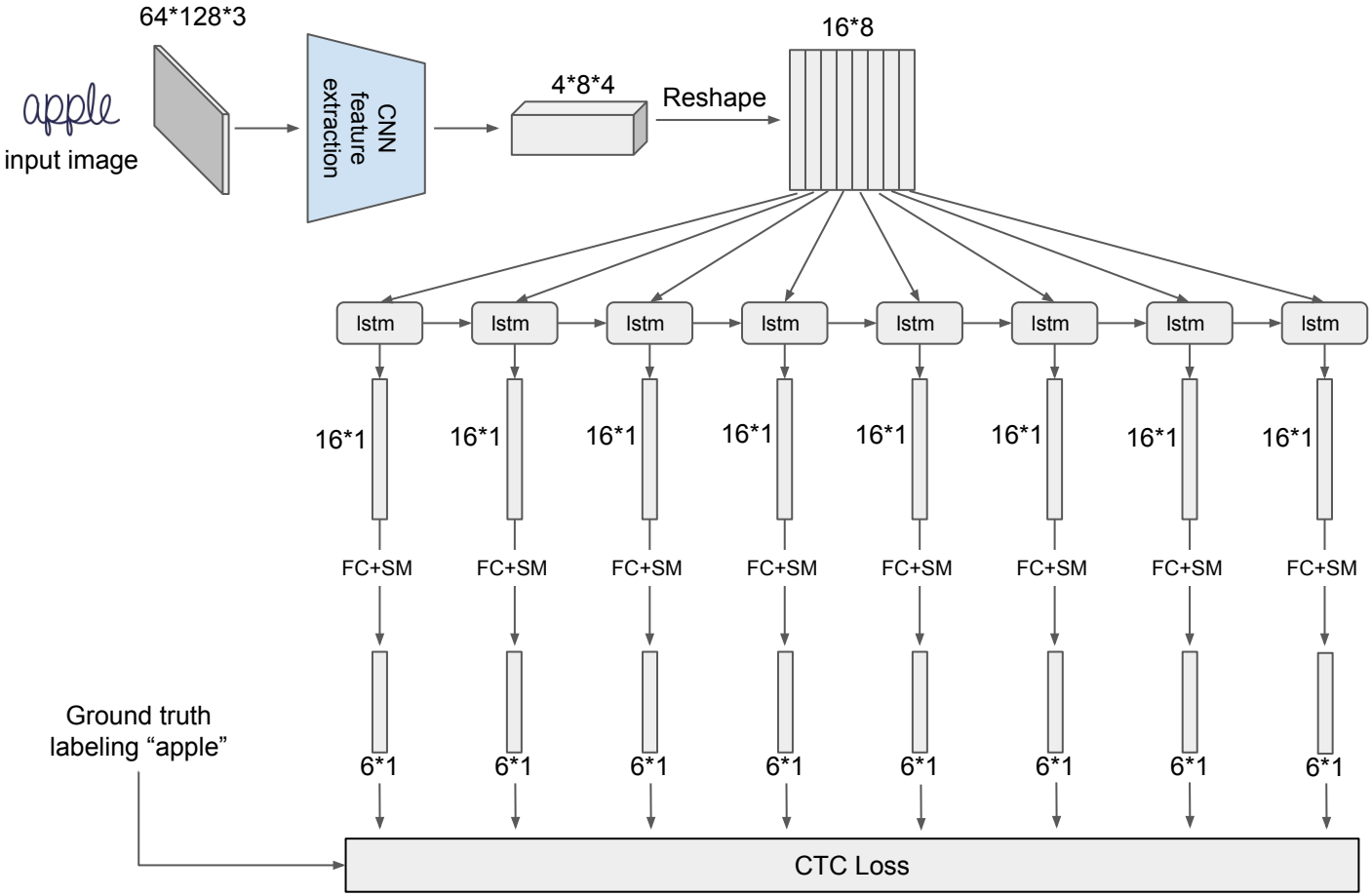

64*128*3
input image
CNN feature extraction
4*8*4
Reshape
16*8
lstm
16*1
FC+SM
6*1
Decoding algorithm
“apple”

Image OCR: model architecture

Training: CTC Loss


64*128*3
apple
input image
CNN feature extraction
4*8*4
Reshape
16*8
lstm
lstm
lstm
lstm
lstm
lstm
lstm
lstm
16*1
16*1
16*1
16*1
16*1
16*1
16*1
16*1
FC+SM
FC+SM
FC+SM
FC+SM
FC+SM
FC+SM
FC+SM
FC+SM
6*1
6*1
6*1
6*1
6*1
6*1
6*1
6*1
Ground truth labeling “apple”
CTC Loss

Thank you

Our Website:

deepsystems.ai

Products:

supervise.ly - Dataset management, annotation and preparation service

movix.ai - Interactive, lstm-based movie recommender system

Outsource projects:

Our team is looking for business partners to make exciting deep learning solutions.