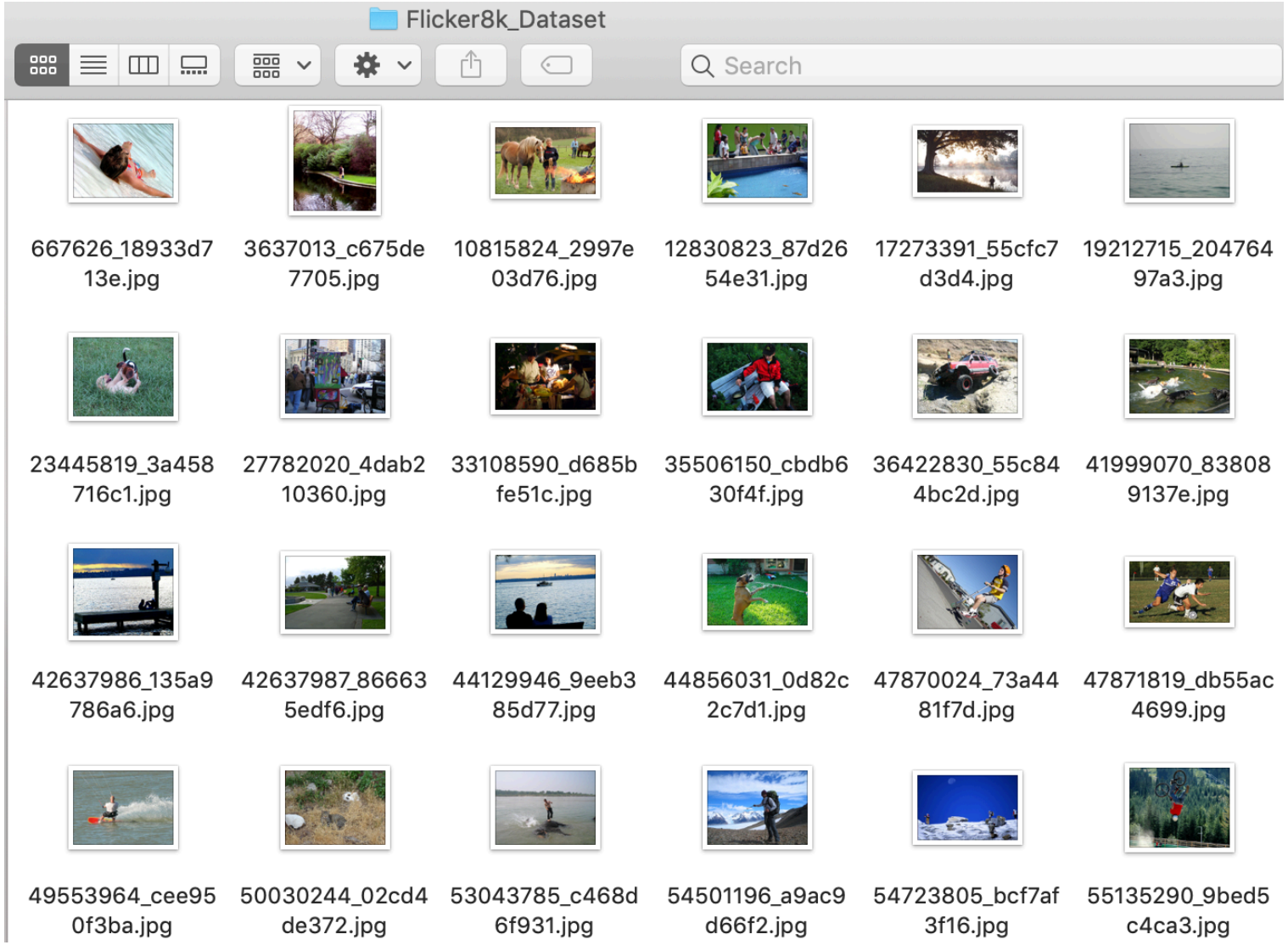
Flickr8K Image Caption Dataset

Image ID: `1000268201_693b08cb0e`

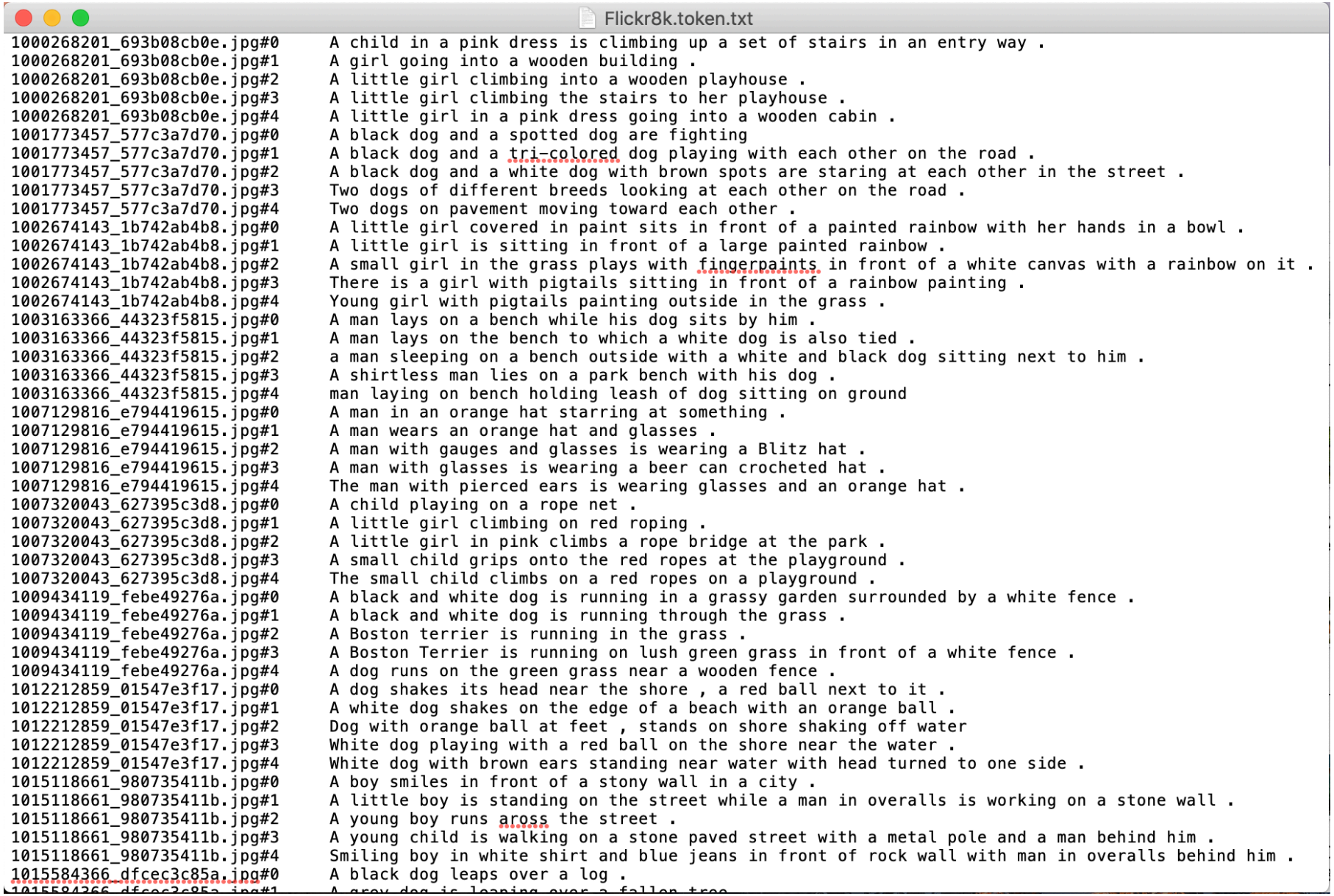
Captions

- A child in a pink dress is climbing up a set of stairs in an entry way
- A girl going into a wooden building
- A little girl climbing into a wooden playhouse
- A little girl climbing the stairs to her playhouse
- A little girl in a pink dress going into a wooden cabin

Flickr8K Image Caption Dataset



Flickr8K Image Caption Dataset



Feature Extraction Using Pretrained VGG16

Pretrained VGG16

```
from keras.applications import VGG16
from keras.models import Model

vgg16 = VGG16(weights='imagenet',
              include_top=True,
              input_shape=(224, 224, 3))

vgg16 = Model(inputs=vgg16.inputs,
              outputs=vgg16.layers[-2].output)

vgg16.summary()
```

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312

Total params: 134,260,544
Trainable params: 134,260,544
Non-trainable params: 0

Pretrained VGG16

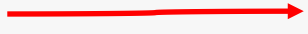
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
⋮	⋮	⋮
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312

Total params: 134,260,544
Trainable params: 134,260,544
Non-trainable params: 0

Feature Extraction

Convert 224×224×3 images to 4096-dim vectors

```
from keras.preprocessing import image
from keras.applications.vgg16 import preprocess_input

img_path = 'Flicker8k_Dataset/667626_18933d713e.jpg'

img = image.load_img(img_path, target_size=(224, 224))
x = image.img_to_array(img)
x = preprocess_input(x)
x = x.reshape(1, 224, 224, 3)

features = vgg16.predict(x)
```



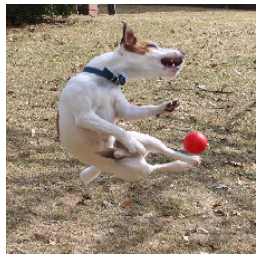
4096-dim vector

Feature Extraction

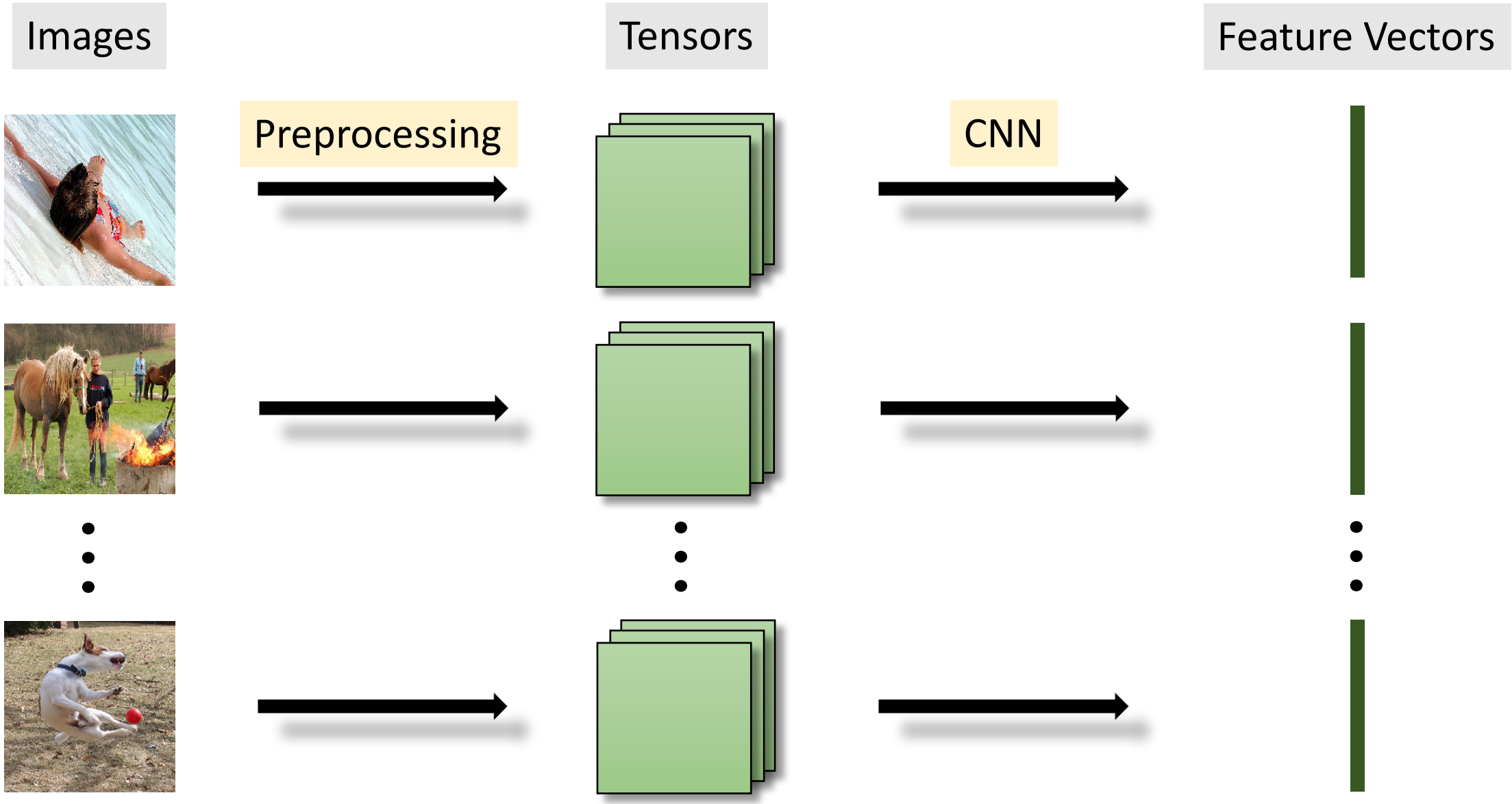
Images

Tensors

Preprocessing



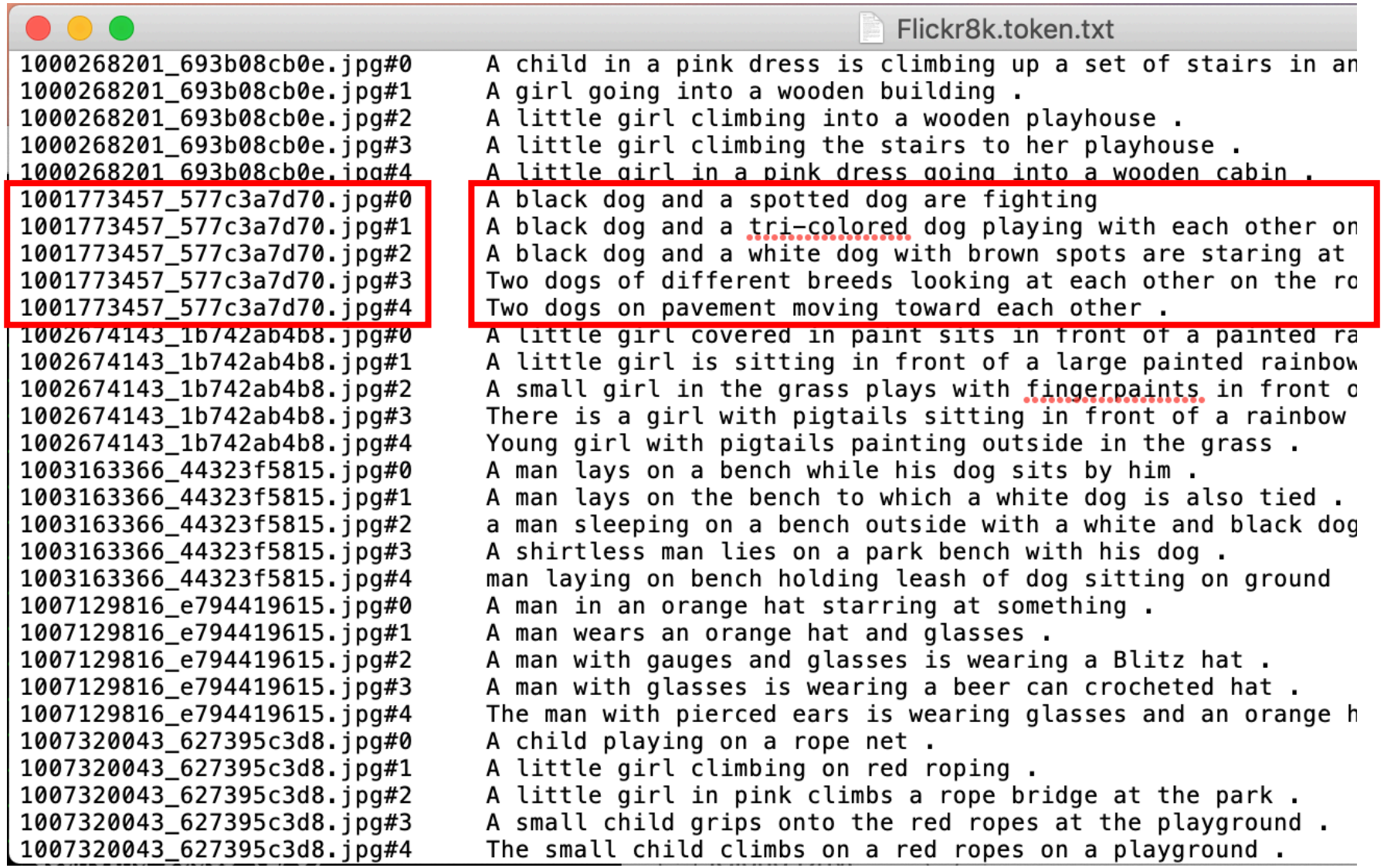
Feature Extraction



Text Processing

Processing Text Data

5 different captions for the same image



Processing Text Data

Convert the file "`Flickr8k.token`" to a list of (ID, caption)

- `('1000268201_693b08cb0e', 'a child in a pink dress is climbing up a set of stairs in an entry way'),`
- `('1000268201_693b08cb0e', 'a girl going into a wooden building'),`
- `('1000268201_693b08cb0e', 'a little girl climbing into a wooden playhouse'),`
- `('1000268201_693b08cb0e', 'a little girl climbing the stairs to her playhouse'),`
- `('1000268201_693b08cb0e', 'a little girl in a pink dress going into a wooden cabin')`
- `('1001773457_577c3a7d70', 'a black dog and a spotted dog are fighting'),`
- `('1001773457_577c3a7d70', 'a black dog and a tri-colored dog playing with each other on the road'),`

⋮

Texts to Sequences

`texts[i]:`

```
'a child in a pink dress is climbing up a set of stairs in an entry way'
```

Tokenization

`tokens[i]:`

```
['startseq', 'a', ' child', 'in', 'a', 'pink', 'dress', 'is', 'climbing', 'up', 'a', 'set', 'of', 'stairs', 'in', 'an', 'entry', 'way', 'endseq']
```

Encoding

`seqs[i]:`

```
[2, 1, 43, 4, 1, 90, 172, 7, 119, 51, 1, 394, 12, 395, 4, 28, 5159, 670, 3]
```

Processing Text Data

Result: a list of (ID, sequence)

- ('1000268201_693b08cb0e', [2, 1, 43, 4, 1, 90, 172, 7, 119, 51, 1, 394, 12, 395, 4, 28, 5159, 670, 3]),
- ('1000268201_693b08cb0e', [2, 1, 19, 316, 64, 1, 196, 117, 3]),
- ('1000268201_693b08cb0e', [2, 1, 40, 19, 119, 64, 1, 196, 2437, 3]),
- ('1000268201_693b08cb0e', [2, 1, 40, 19, 119, 5, 395, 20, 60, 2437, 3]),
- ('1000268201_693b08cb0e', [2, 1, 40, 19, 4, 1, 90, 172, 316, 64, 1, 196, 2981, 3]),
- ('1001773457_577c3a7d70', [2, 1, 15, 9, 8, 1, 843, 9, 17, 343, 3]),
- ('1001773457_577c3a7d70', [2, 1, 15, 9, 8, 1, 1575, 235, 9, 34, 10, 137, 82, 6, 5, 151, 3])

⋮

Prepare Training Data

Training Data

- Inputs: (**image feature**, **sequence**)
- Target: **next word**

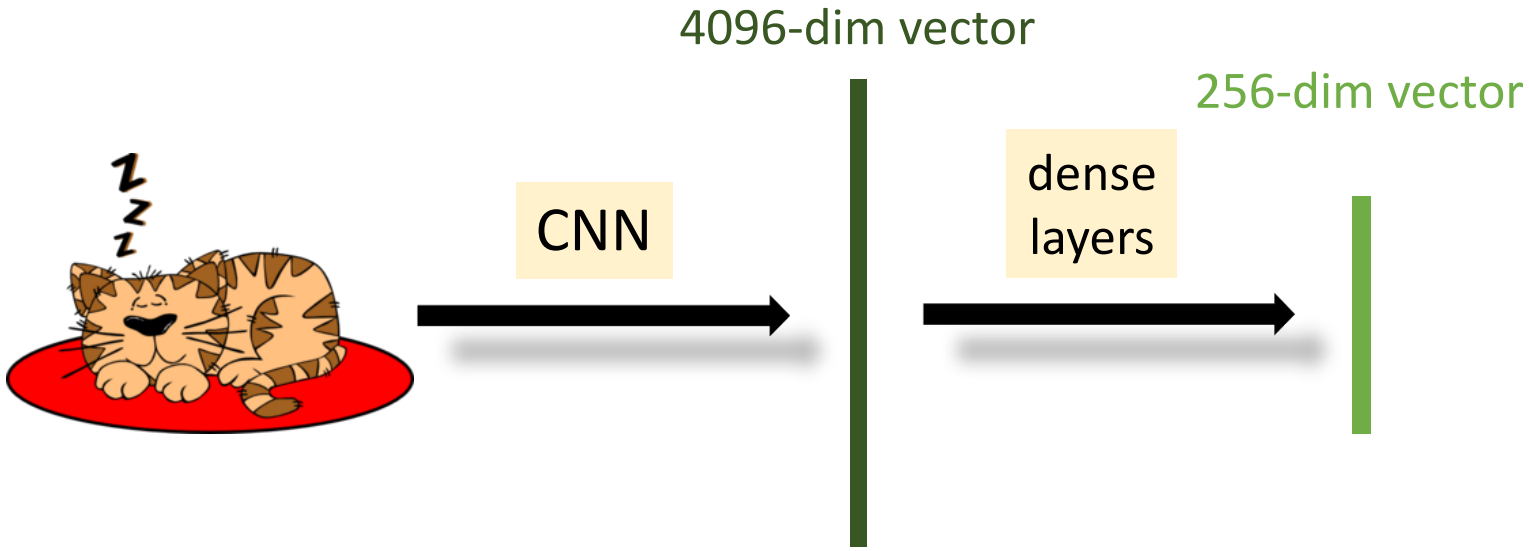
x1	x2	y
image feature	'startseq'	'a'
image feature	'startseq a'	'cat'
image feature	'startseq a cat'	'sat'
image feature	'startseq a cat sat'	'on'
image feature	'startseq a cat sat on'	'a'
image feature	'startseq a cat sat on a'	'mat'
image feature	'startseq a cat sat on a mat'	'endseq'

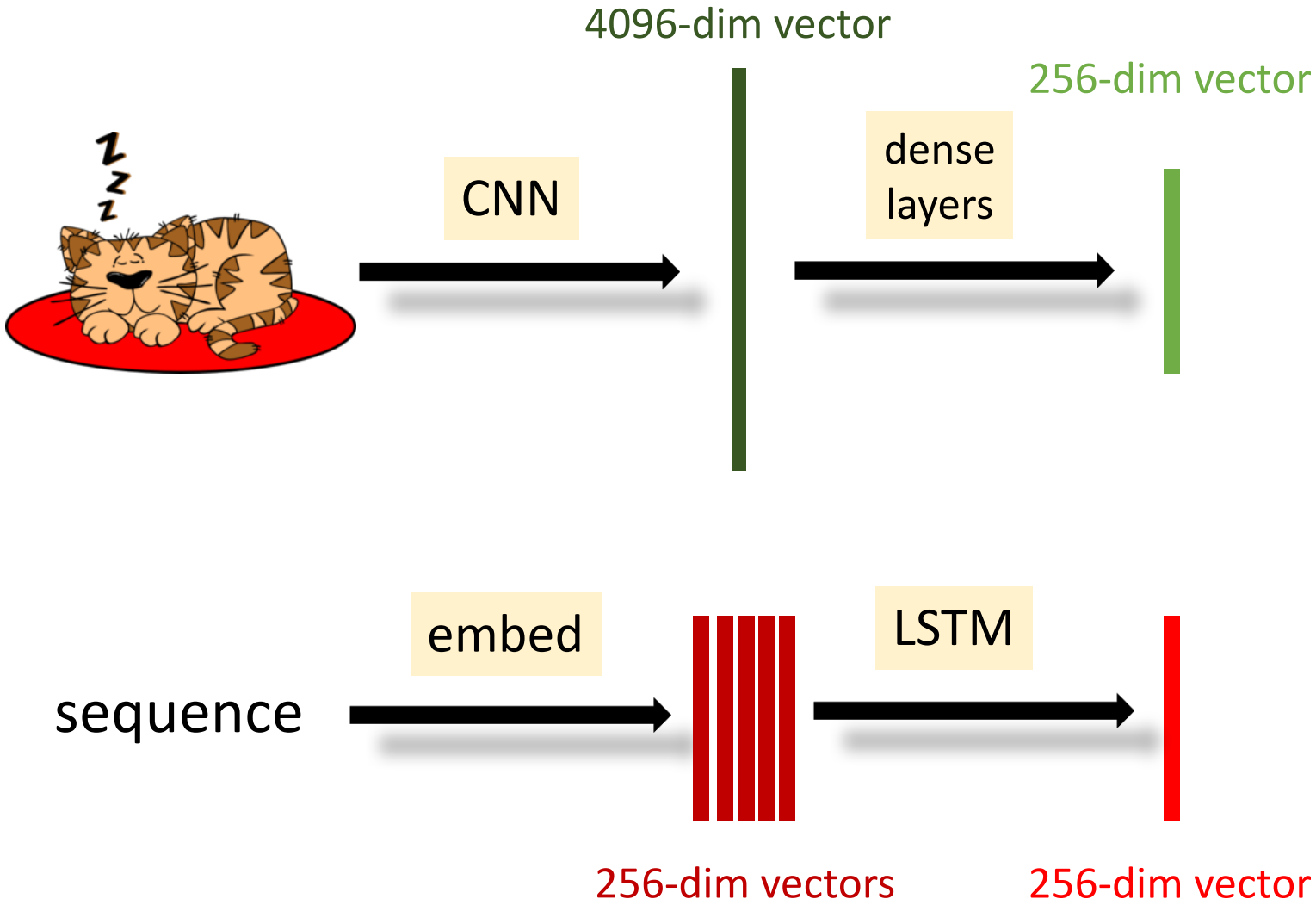
Training Data

- Inputs: (image feature, sequence)
- Target: next word

- #training images: 6K
- #captions: 30K
- #(x1, x2, y) triplets: 354K

Deep Learning Model


4096-dim vector
256-dim vector
CNN
dense
layers


4096-dim vector
256-dim vector
CNN
dense layers
sequence
embed
LSTM
256-dim vectors
256-dim vector

4096-dim vector

CNN

dense layers

256-dim vector

add or **concatenate**

256-dim vector or 512-dim vector

sequence

embed

LSTM

256-dim vectors

256-dim vector

4096-dim vector

CNN

dense layers

256-dim vector

add or **concatenate**

dense layers

sequence

embed

LSTM

256-dim vectors

256-dim vector

256-dim vector or 512-dim vector

8496-dim vector

(vocabulary size = 8496)

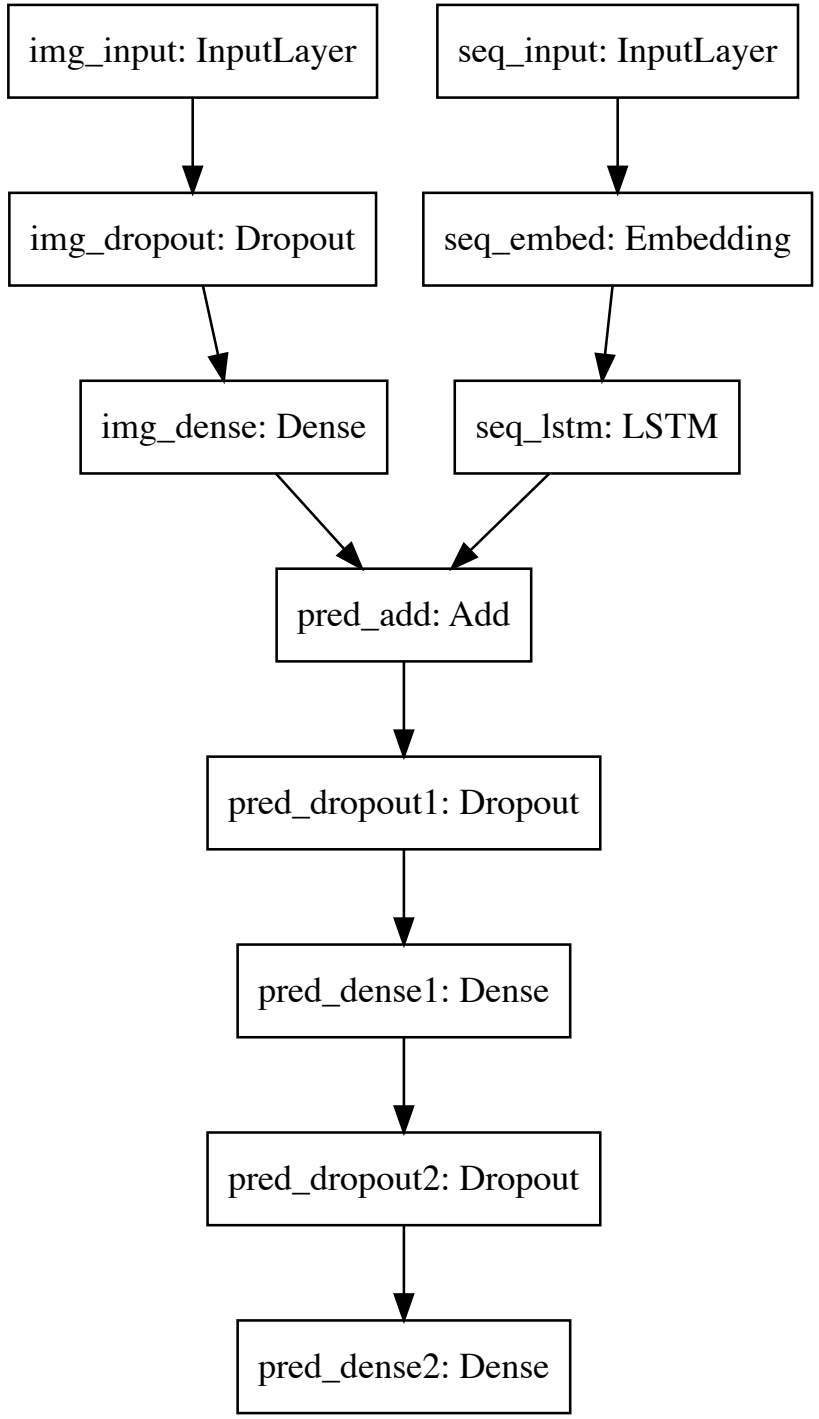
Network Structure

```
from keras.layers import Input, Dropout, Dense, Embedding, LSTM, Flatten, Add

img_input = Input(shape=(4096,), name='img_input')
img_dropout = Dropout(0.5, name='img_dropout')(img_input)
img_dense = Dense(256, activation='relu', name='img_dense')(img_dropout)

seq_input = Input(shape=(max_len,), name='seq_input')
seq_embed = Embedding(vocabulary, 256, name='seq_embed')(seq_input)
seq_lstm = LSTM(256, dropout=0.2, name='seq_lstm')(seq_embed)

pred_add = Add(name='pred_add')([img_dense, seq_lstm])
pred_dropout1 = Dropout(0.5, name='pred_dropout1')(pred_add)
pred_dense1 = Dense(256, activation='relu', name='pred_dense1')(pred_dropout1)
pred_dropout2 = Dropout(0.5, name='pred_dropout2')(pred_dense1)
outputs = Dense(vocabulary, activation='softmax', name='pred_dense2')(pred_dropout2)
```

Network Structure

```
from keras.models import Model
model = Model(inputs=[img_input, seq_input], outputs=outputs)

model.summary()
```

```
_________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================
img_input (InputLayer)       (None, 4096)              0
_________________________________________________________________________________
seq_input (InputLayer)       (None, 39)                0
_________________________________________________________________________________
img_dropout (Dropout)        (None, 4096)              0           img_input[0][0]
_________________________________________________________________________________
seq_embed (Embedding)        (None, 39, 256)           2174976     seq_input[0][0]
_________________________________________________________________________________
img_dense (Dense)            (None, 256)               1048832     img_dropout[0][0]
_________________________________________________________________________________
seq_lstm (LSTM)              (None, 256)               525312      seq_embed[0][0]
_________________________________________________________________________________
pred_add (Add)               (None, 256)               0           img_dense[0][0]
                                                                   seq_lstm[0][0]
_________________________________________________________________________________
pred_dropout1 (Dropout)      (None, 256)               0           pred_add[0][0]
_________________________________________________________________________________
pred_dense1 (Dense)          (None, 256)               65792       pred_dropout1[0][0]
_________________________________________________________________________________
pred_dropout2 (Dropout)      (None, 256)               0           pred_dense1[0][0]
_________________________________________________________________________________
pred_dense2 (Dense)          (None, 8496)              2183472     pred_dropout2[0][0]
=================================================================================
Total params: 5,998,384
Trainable params: 5,998,384
Non-trainable params: 0
```



Train the Model

```
model.compile(loss='categorical_crossentropy',
              optimizer='rmsprop')
```

images' features

sequence

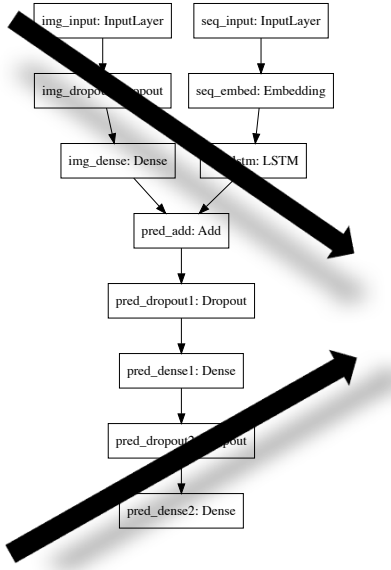
target word

```
model.fit([x1_train, x2_train], y_train,
          batch_size=32, epochs=20, validation_split=0.2)
```

Generate Caption

Make Predictions

Input Image



'startseq'

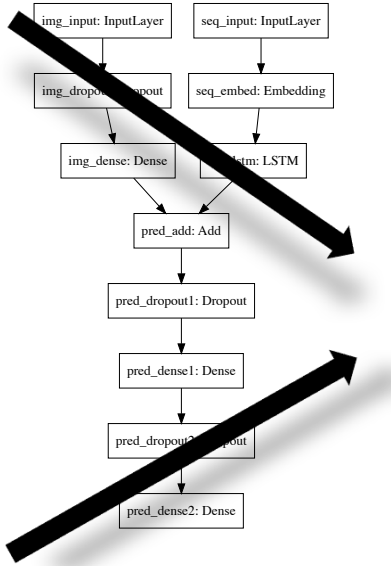
Input Sequence

sampling

'a'

Make Predictions

Input Image



'startseq a'

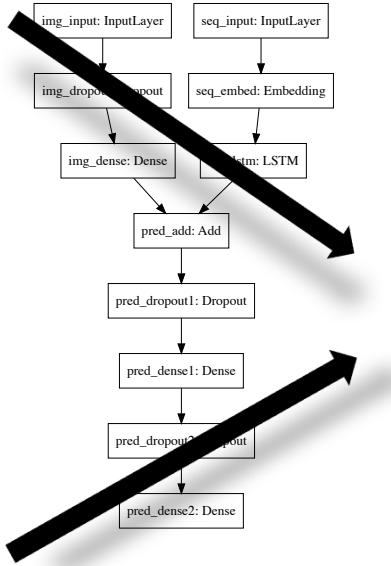
sampling

'cat'

Input Sequence

Make Predictions

Input Image



'startseq a cat'

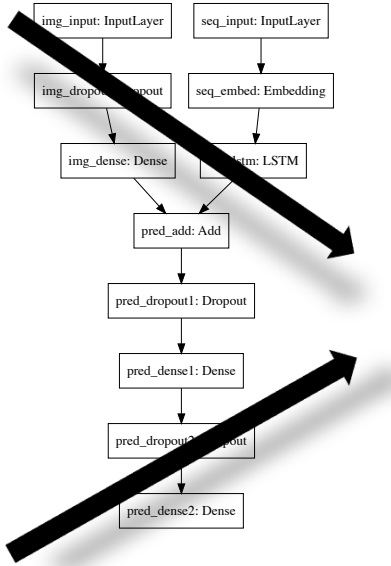
Input Sequence

sampling

'sat'

Make Predictions

Input Image



'startseq a cat sat'

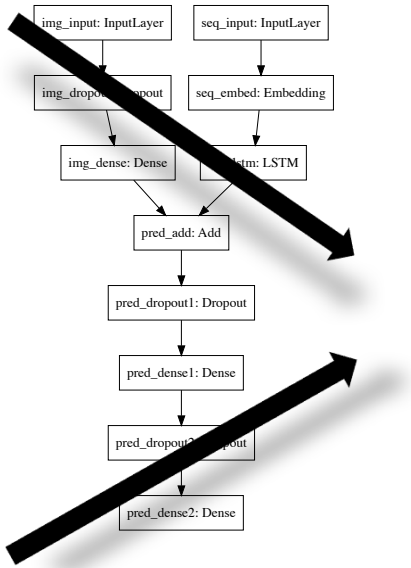
Input Sequence

sampling

'on'

Make Predictions

Input Image



'startseq a cat sat on'

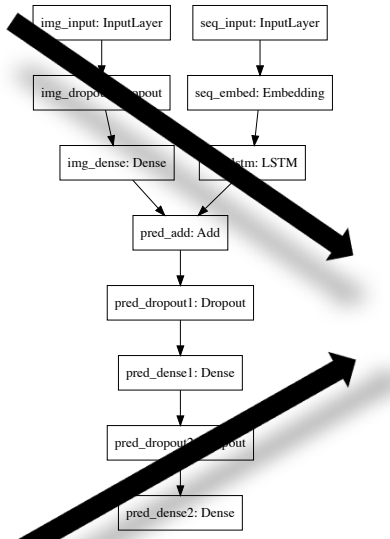
Input Sequence

sampling

'the'

Make Predictions

Input Image



sampling

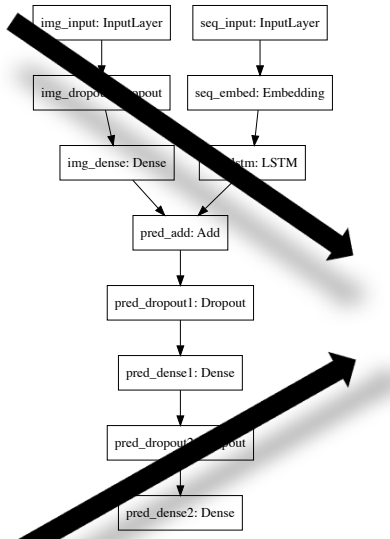
'mat'

'startseq a cat sat on the'

Input Sequence

Make Predictions

Input Image



sampling

'endseq'

'startseq a cat sat on the mat'

Input Sequence

Details

Read the hand-on tutorial:

- https://machinelearningmastery.com/develop-a-deep-learning-caption-generation-model-in-python/