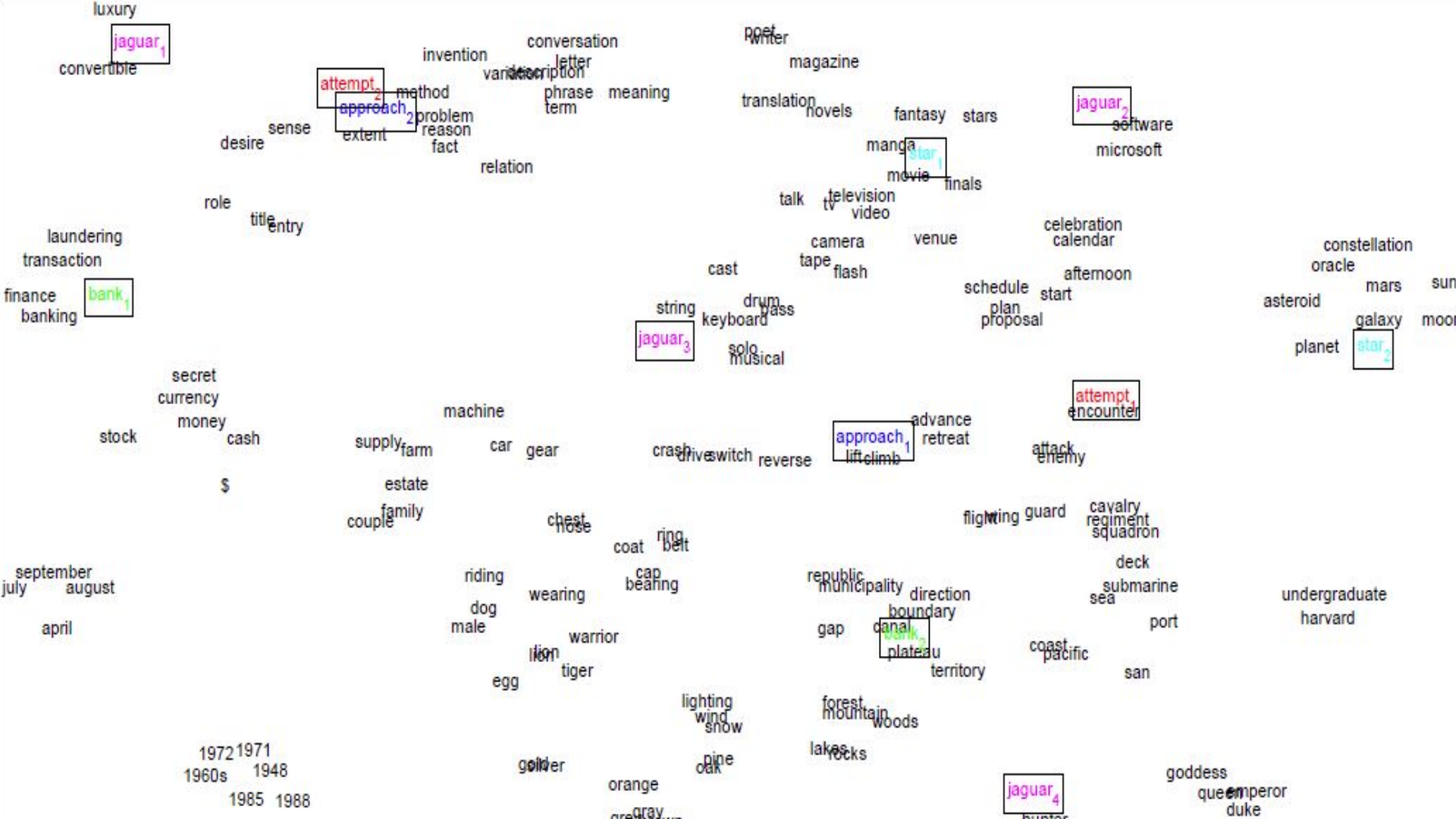

luxury
jaguar1
convertible
attempt2
approach2
invention
conversation
letter
description
method
phrase
meaning
problem
reason
fact
sense
extent
desire
relation
term
poet
writer
magazine
translation
novels
fantasy
stars
jaguar2
software
microsoft
manga
star1
movie
finals
talk
tv
television
video
role
title
entry
laundering
transaction
finance
bank1
banking
camera
venue
celebration
calendar
cast
tape
flash
afternoon
schedule
start
plan
proposal
string
drum
bass
keyboard
jaguar3
solo
musical
constellation
oracle
mars
asteroid
galaxy
planet
star2
secret
currency
money
stock
cash
machine
supply
farm
car
gear
crash
drive
switch
reverse
approach1
lift
climb
advance
retreat
attack
enemy
attempt1
encounter
$
estate
family
couple
chest
nose
ring
coat
belt
flight
wing
guard
cavalry
regiment
squadron
deck
september
july
august
april
riding
cap
bearing
wearing
dog
male
warrior
lion
tiger
egg
republic
municipality
direction
boundary
gap
canal
bank2
plateau
sea
submarine
undergraduate
harvard
port
coast
pacific
territory
san
lighting
wind
snow
forest
mountain
woods
lakes
rocks
pine
oak
gold
orange
gray
1972
1971
1960s
1948
1985
1988
jaguar4
goddess
queen
emperor
duke

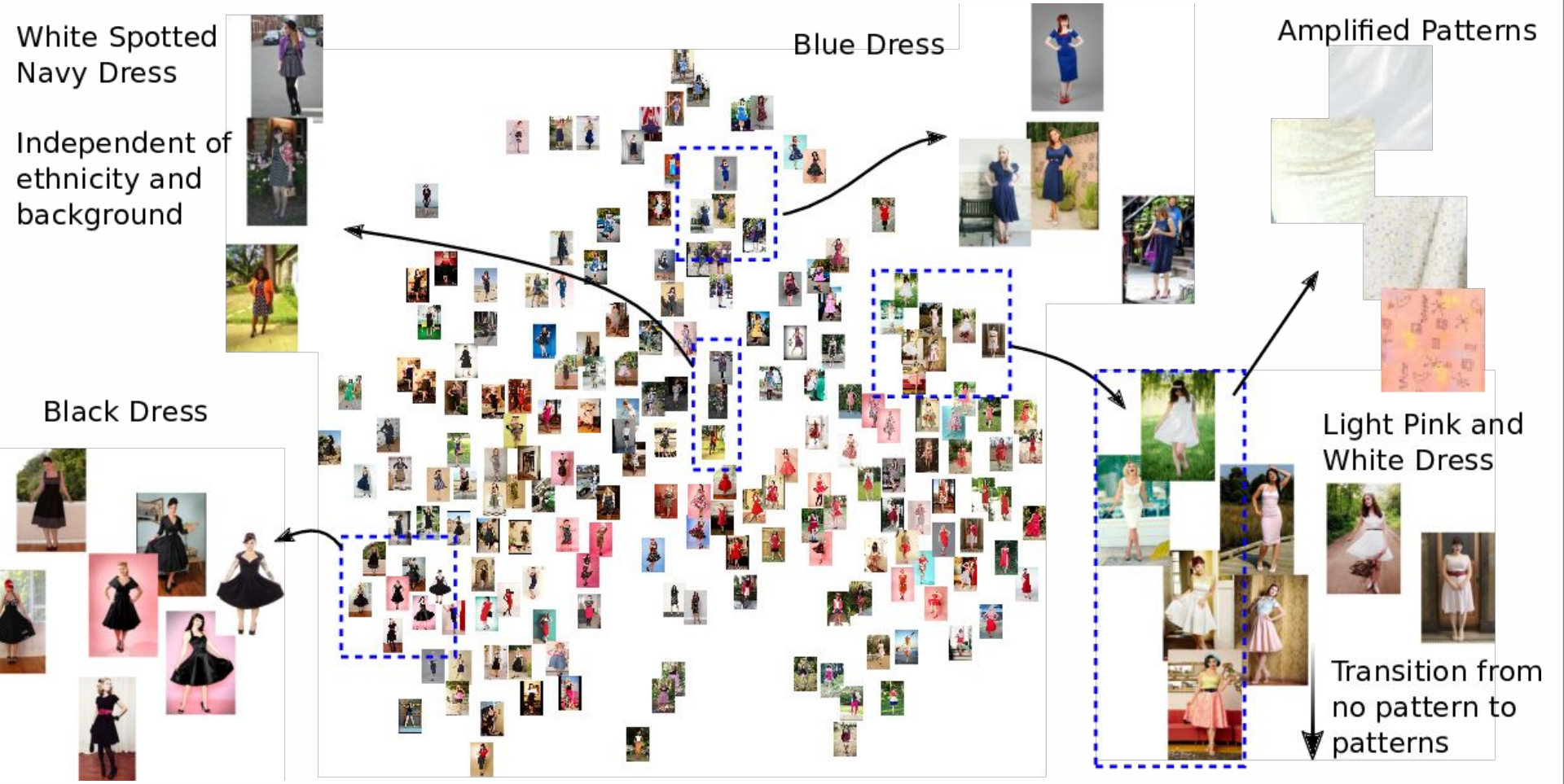

White Spotted Navy Dress
Independent of ethnicity and background
Blue Dress
Amplified Patterns
Black Dress
Light Pink and White Dress
Transition from no pattern to patterns

Image Captioning Based on Deep Neural Networks

TABLE OF CONTENTS

INTRODUCTION

- Image captioning is a task that a machine learns to generate natural language sentences to describe the salient parts of an image.
- A description must capture not only the objects contained in an image, but it also must express how these objects relate to each other as well as their attributes and the activities they are involved in.
- Thus, accurate image captioning is a challenging task that requires advancing the state of the art of both computer vision and natural language processing.

CAN YOU WRITE A CAPTION?

02

METHODOLOGY

The main inspiration of our work comes from recent advances in machine translation, where the task is to transform a sentence S written in a source language, into its translation T in the target language, by maximizing P(T|S).

Machine Translation Architecture

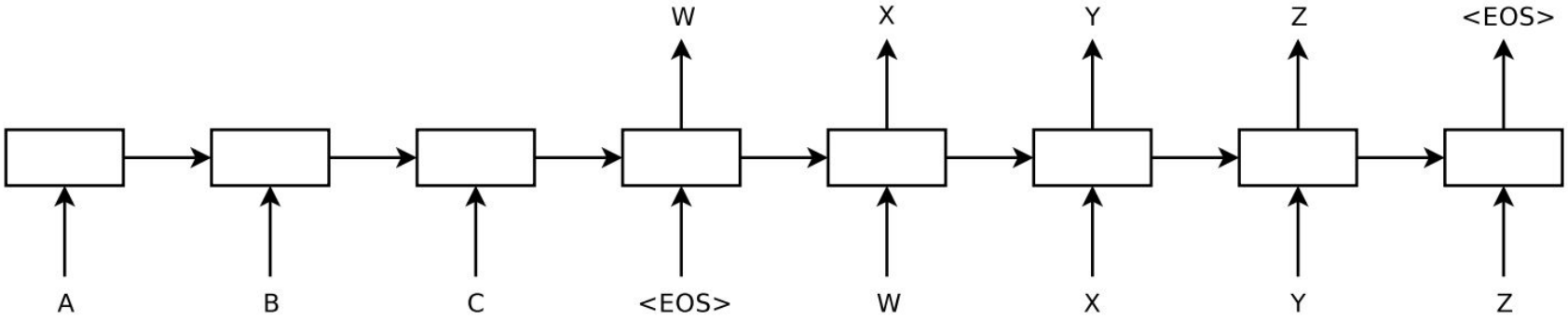


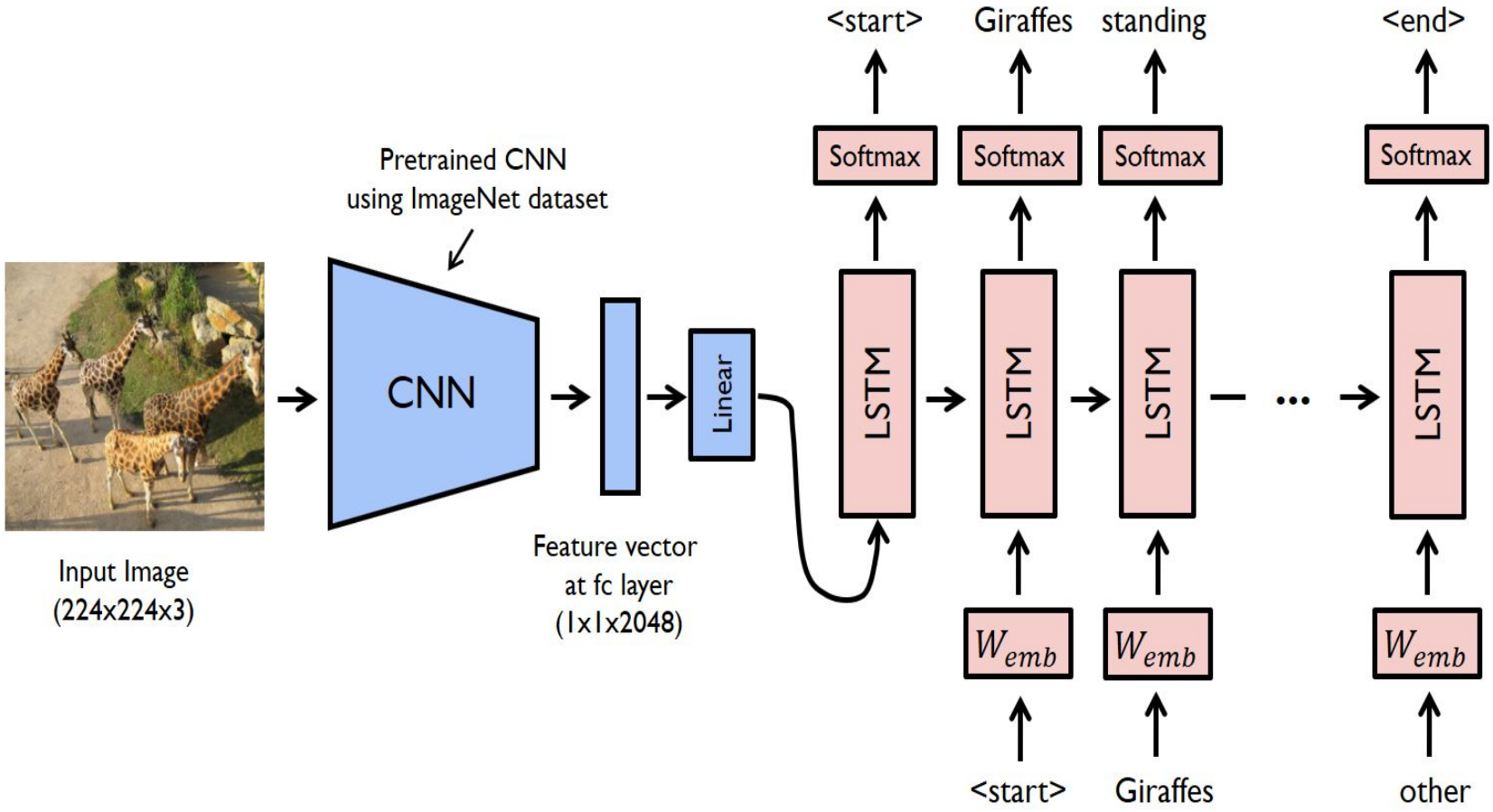
IMAGE-TO-TEXT TRANSLATION IS OBTAINED BY FOLLOWING TWO STEPS

ENCODER CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign learnable weights and biases to various aspects/objects in the image and be able to differentiate one from the other. An deep encoder CNN will produce a rich representation of input by embedding it in a fixed-length vector.

DECODER RECURRENT NEURAL NETWORK

A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. A decoder RNN uses the last hidden layer as an input to the RNN decoder which uses the fixed dimensional vector representation to “decode” it to the desired output sentence.


<start>
Giraffes
standing
<end>
Softmax
Softmax
Softmax
Softmax
Pretrained CNN
using ImageNet dataset
CNN
Linear
LSTM
LSTM
LSTM
LSTM
W_{emb}
W_{emb}
W_{emb}
<start>
Giraffes
other
Input Image
(224x224x3)
Feature vector
at fc layer
(1x1x2048)

WHAT WE HAVE LEARNED SO FAR

01 Neural Network

02 Convolutional Neural Network

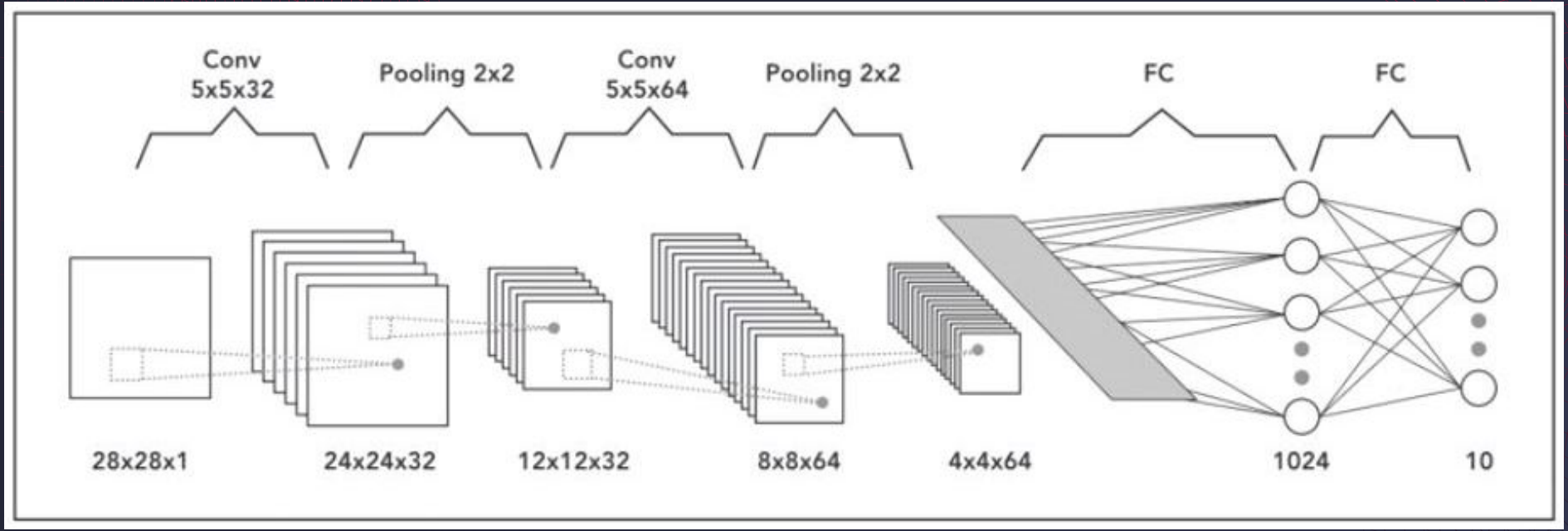
03 Mathematics of Recurrent Neural Network

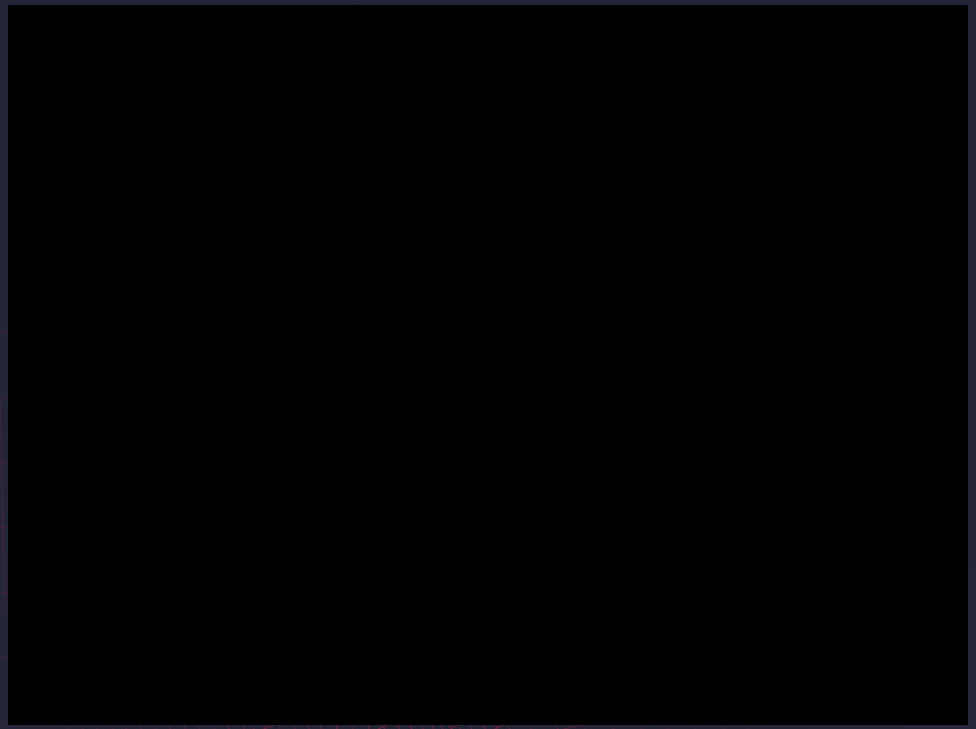
04 Pre-trained Models in Keras

05 GRU and LSTM

CONVOLUTIONAL NEURAL NETWORK

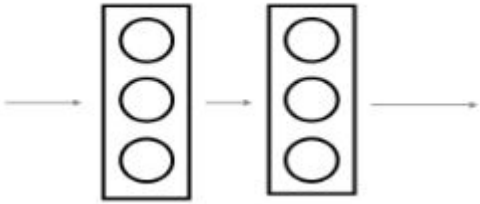
- CNNs, like neural networks, are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, pass it through an activation function and responds with an output.
- A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and normalization layers.
- It preserves the relationship between pixels by learning image features using small squares of input data.



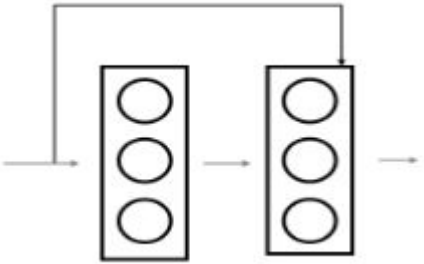


BUILDING A RESNET

Without Skip Connection



With Skip Connection



- In ResNets, a "shortcut" or a "skip connection" allows the gradient to be directly back propagated to earlier layers.
- Two main types of blocks are used in a ResNet, depending mainly on whether the input/output dimensions are same or different.

    - The Identity Block

The identity block is the standard block used in ResNets, and corresponds to the case where the input activation (say $a[l]$) has the same dimension as the output activation (say $a[l+2]$).

    - The Convolution Block

This type of block is used when the input and output dimensions don't match up. The difference with the identity block is that there is a CONV2D layer in the shortcut path:

THE IDENTITY BLOCK

First component of main path:

- The first CONV2D has $F1$ filters of shape (1,1) and a stride of (1,1).Its padding is "valid" .
- The first BatchNorm is normalizing the channels axis.
- Then apply the ReLU activation function. This has no name and no hyperparameters.

Second component of main path:

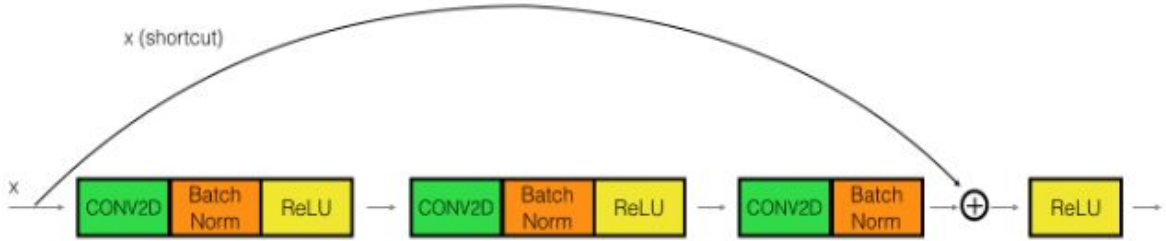
- The second CONV2D has $F2$ filters of shape (f,f) and a stride of (1,1).Its padding is "same" .
- The second BatchNorm is normalizing the channels axis.
- Then apply the ReLU activation function. This has no name and no hyperparameters.

Third component of main path:

- The third CONV2D has $F3$ filters of shape (1,1) and a stride of (1,1). Its padding is "valid".
- The third BatchNorm is normalizing the channels axis. Note that there is no ReLU activation function in this component.

Final step:

- The shortcut and the input are added together.
- Then apply the ReLU activation function. This has no name and no hyperparameters.



THE CONVOLUTION BLOCK

First component of main path:

- The first CONV2D has $F1$ filters of shape (1,1) and a **stride of (s,s)**. Its padding is "valid".
- The first BatchNorm is normalizing the channels axis.
- Then apply the ReLU activation function. This has no name and no hyperparameters.

Second component of main path:

- The second CONV2D has $F2$ filters of (f,f) and a stride of (1,1). Its padding is "same".
- The second BatchNorm is normalizing the channels axis.
- Then apply the ReLU activation function. This has no name and no hyperparameters.

Third component of main path:

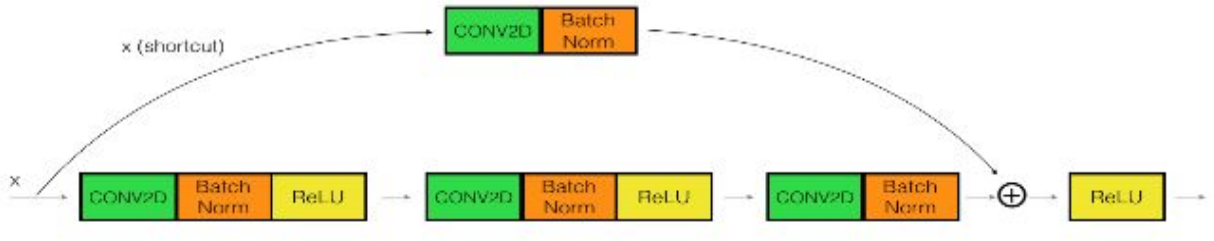
- The third CONV2D has $F3$ filters of (1,1) and a stride of (1,1). Its padding is "valid".
- The third BatchNorm is normalizing the channels axis. Note that there is no ReLU activation function in this component.

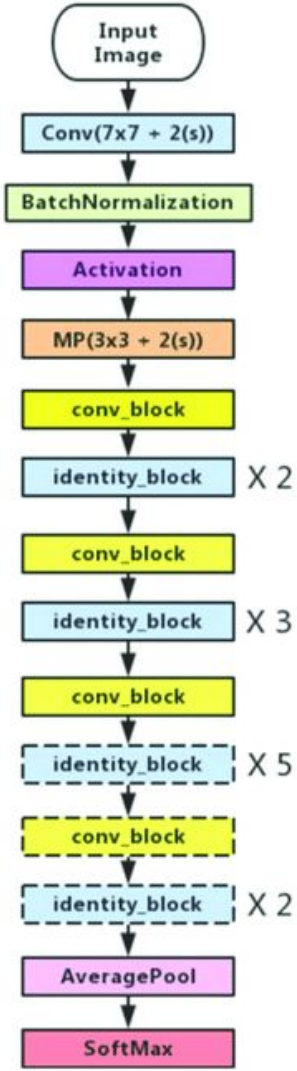
Shortcut path:

- The CONV2D has $F3$ filters of shape (1,1) and a stride of (s,s). Its padding is "valid".
- The BatchNorm is normalizing the channels axis.

Final step:

- The shortcut and the main path values are added together.
- Then apply the ReLU activation function. This has no name and no hyperparameters.





ARCHITECTURE OF RESNET

The details of this ResNet-50 model are:

- Zero-padding pads the input with a pad of (3,3)

- Stage 1:
  - The 2D Convolution has 64 filters of shape (7,7) and uses a stride of (2,2).
  - BatchNorm is applied to the channels axis of the input.
  - Max Pooling uses a (3,3) window and a (2,2) stride.

- Stage 2:
  - The convolutional block uses three set of filters of size [64,64,256], "f" is 3, "s" is 1.
  - The 2 identity blocks use three set of filters of size [64,64,256], "f" is 3.

- Stage 3:
  - The convolutional block uses three set of filters of size [128,128,512], "f" is 3, "s" is 2.
  - The 3 identity blocks use three set of filters of size [128,128,512], "f" is 3.

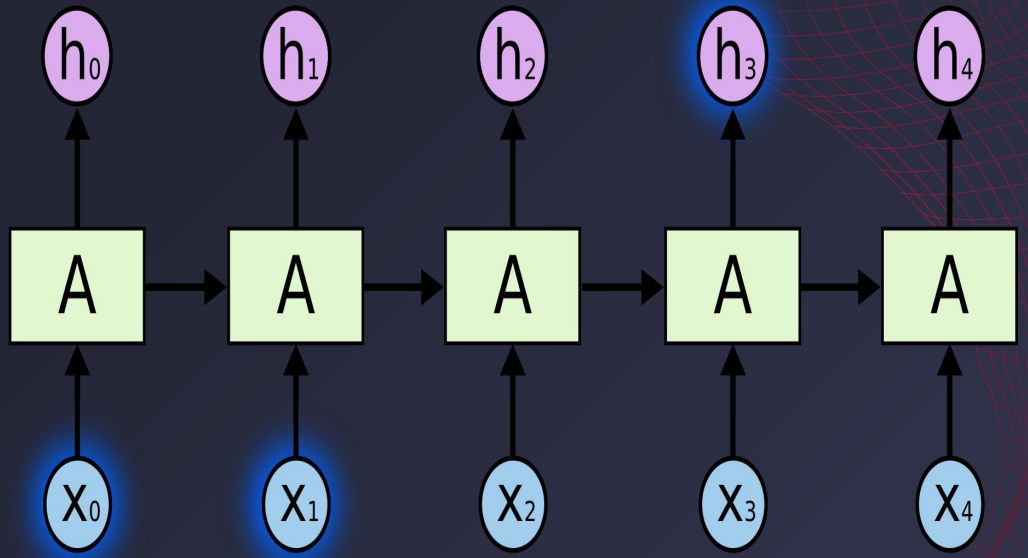
- Stage 4:
  - The convolutional block uses three set of filters of size [256, 256, 1024], "f" is 3, "s" is 2.
  - The 5 identity blocks use three set of filters of size [256, 256, 1024], "f" is 3.

- Stage 5:
  - The convolutional block uses three set of filters of size [512, 512, 2048], "f" is 3, "s" is 2.
  - The 2 identity blocks use three set of filters of size [512, 512, 2048], "f" is 3.

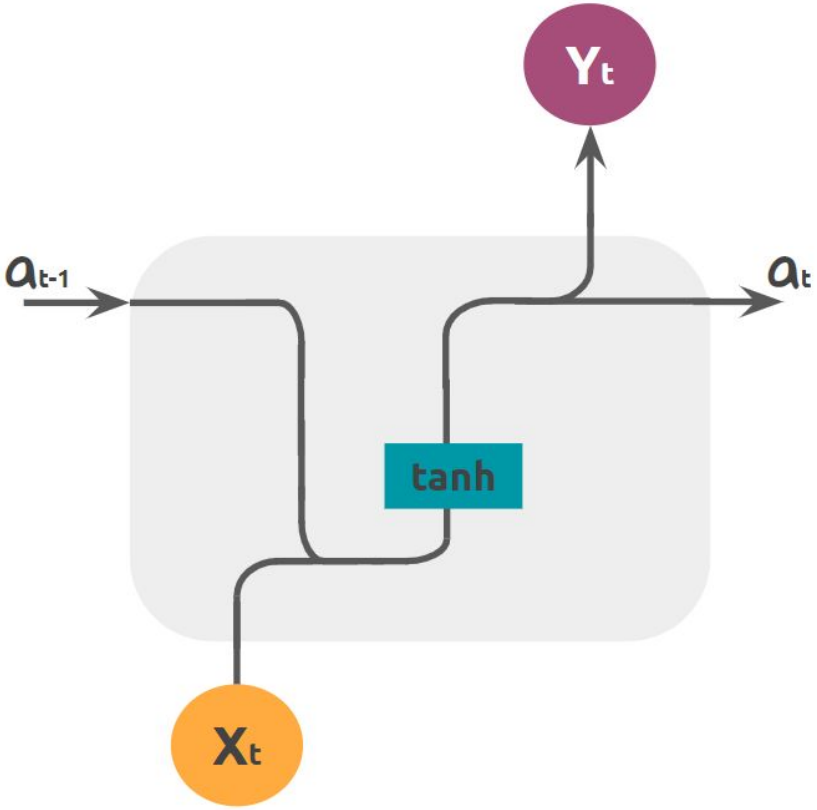
- The 2D Average Pooling uses a window of shape (2,2) and its name is "avg_pool".
- The flatten doesn't have any hyperparameters or name.
- The Fully Connected (Dense) layer reduces its input to the number of classes using a softmax activation.

RECURRENT NEURAL NETWORK

- A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior
- Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture[1] used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).
- A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

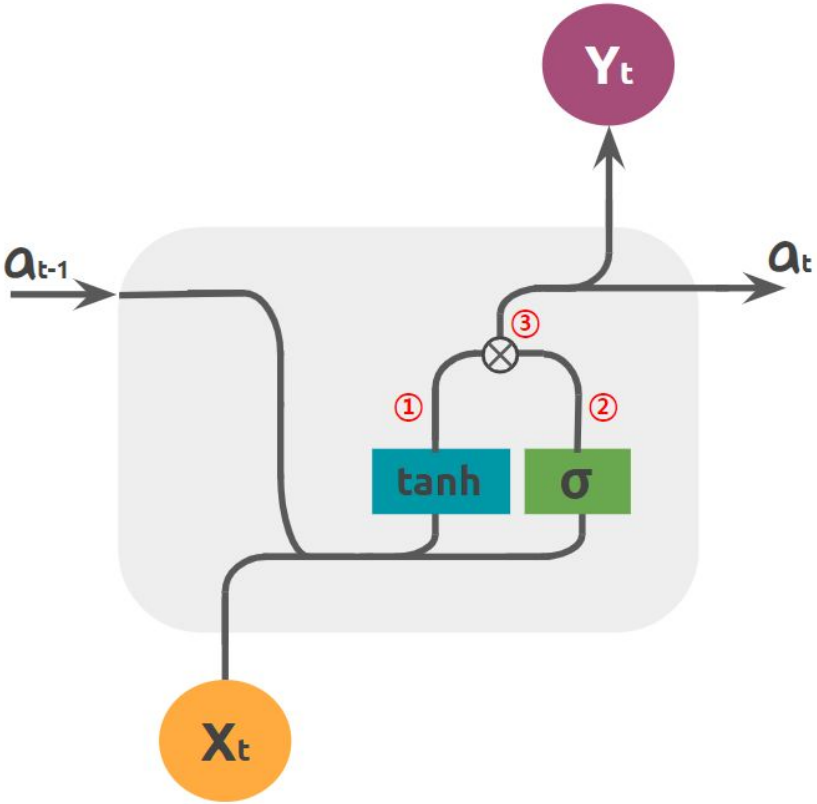


BASIC RNN



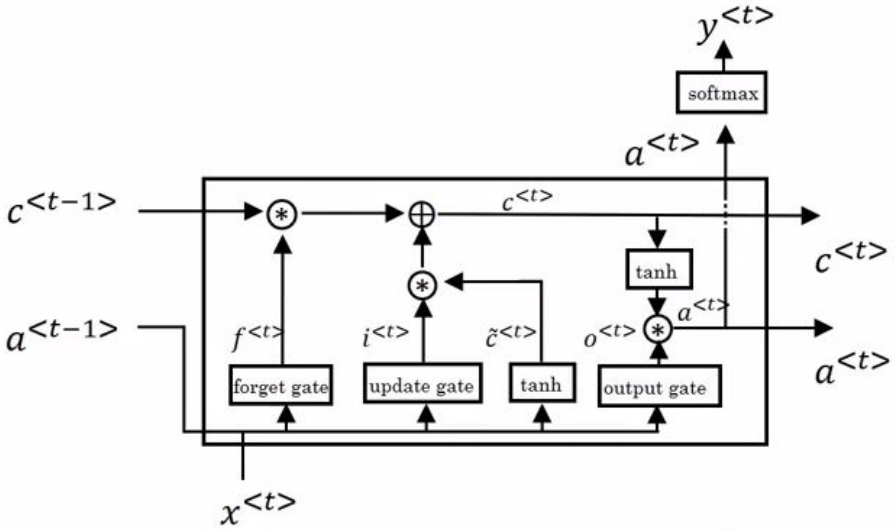
< RNN >

GRU



< GRUs >

LSTM IN A PICTURE



$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$
$$\Gamma_u = \sigma(W_u[\, a^{<t-1>}, x^{<t>}] + b_u)$$
$$\Gamma_f = \sigma(W_f[\, a^{<t-1>}, x^{<t>}] + b_f)$$
$$\Gamma_o = \sigma(W_o[\, a^{<t-1>}, x^{<t>}] + b_o)$$
$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$
$$a^{<t>} = \Gamma_o * \tanh c^{<t>}$$

03
DATA SET

Google’s Conceptual Captioning Dataset

- On September 5, 2018 Google introduced Conceptual Captions, a new dataset consisting of ~3.3 million image/caption pairs that are created by automatically extracting and filtering image caption annotations from billions of web pages.
- As measured by human raters, the machine-curated Conceptual Captions has an accuracy of ~90%.
- Furthermore, because images in Conceptual Captions are pulled from across the web, it represents a wider variety of image-caption styles than previous datasets, allowing for better training of image captioning models.
- To generate this dataset, a Flume pipeline processes billions of Internet web pages, extracting, filtering, and processing candidate image and caption pairs, and keeping those that pass through several filters.
- The filtering and processing steps are as follows :
    - Image-based filtering
    - Text-based filtering
    - Image and Text based filtering
    - Text Transformation with Hypernymization

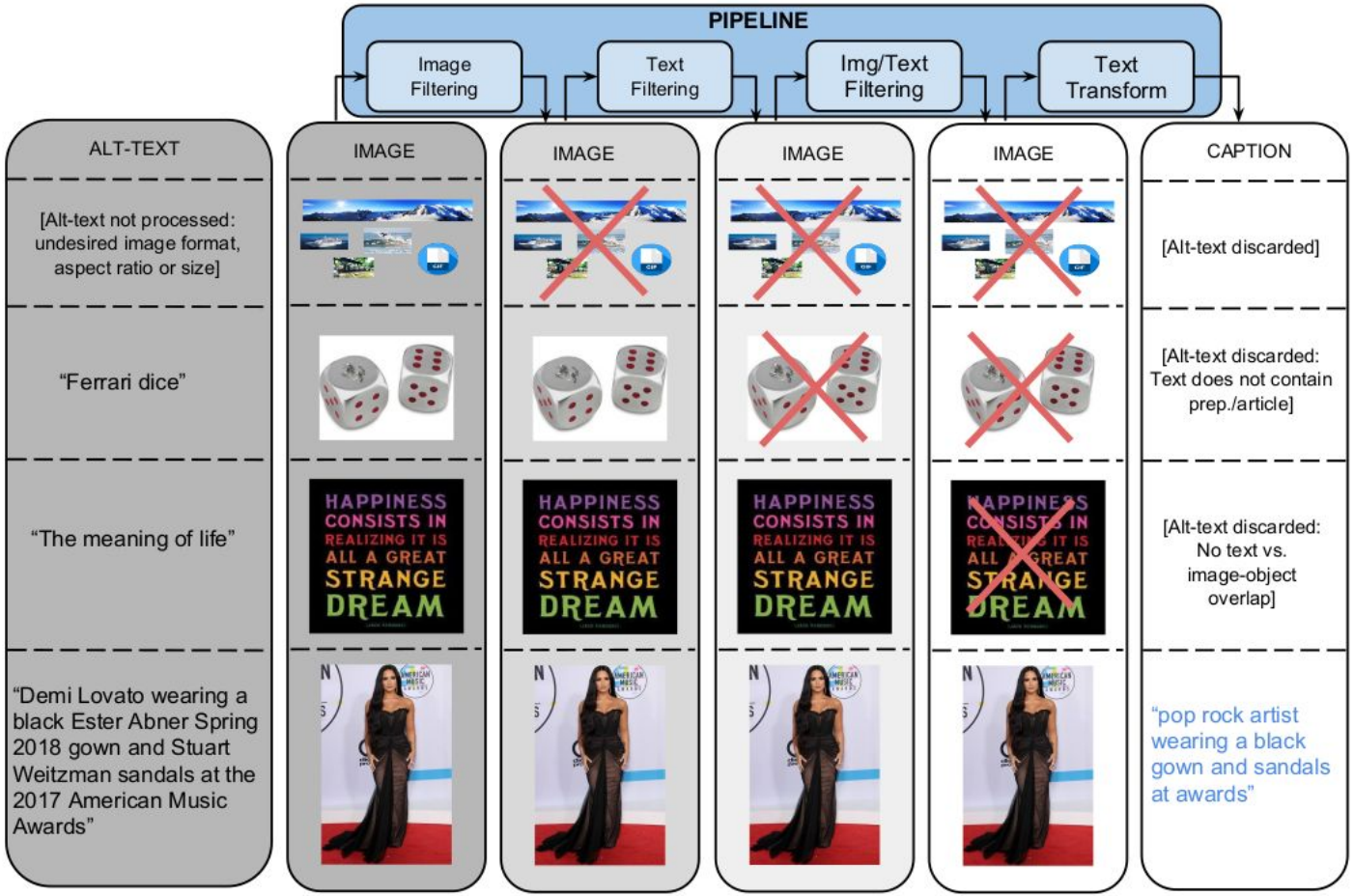

PIPELINE
Image Filtering
Text Filtering
Img/Text Filtering
Text Transform
ALT-TEXT
IMAGE
IMAGE
IMAGE
IMAGE
CAPTION
[Alt-text not processed: undesired image format, aspect ratio or size]
[Alt-text discarded]
“Ferrari dice”
[Alt-text discarded: Text does not contain prep./article]
“The meaning of life”
HAPPINESS CONSISTS IN REALIZING IT IS ALL A GREAT STRANGE DREAM
[Alt-text discarded: No text vs. image-object overlap]
“Demi Lovato wearing a black Ester Abner Spring 2018 gown and Stuart Weitzman sandals at the 2017 American Music Awards”
“pop rock artist wearing a black gown and sandals at awards”

Illustration of Images and captions in Conceptual Captioning Dataset

	1+	2+	3+
Conceptual Captions	96.9%	90.3%	78.5%

"trees in a winter snowstorm"

"a cartoon illustration of a bear waving and smiling"

"the scenic route through mountain range includes these unbelievably coloured mountains"

"facade of an old shop"

DATASET STATS

TRAINING SPLIT

The Training split consists of 3,318,333 image-URL/caption pairs, with a total number of 51,201 total token types in the captions (i.e., total vocabulary). The average number of tokens per captions is 10.3 (standard deviation of 4.5), while the median is 9.0 tokens per caption.

VALIDATION SPLIT

The Validation split consists of 28,355 image-URL/caption pairs, with a total number of 13,063 total token types in the captions (i.e., total vocabulary). The average number of tokens per captions is 10.3 (standard deviation of 4.6), while the median is 9.0 tokens per caption.

TESTING SPLIT

The Testing split consists of 22,530 image-URL/caption pairs, with a total number of 11,731 total token types in the captions (i.e., total vocabulary). The average number of tokens per captions is 10.1 (standard deviation of 4.5), while the median is 9.0 tokens per caption.

04
TECHNOLOGY STACK

TECHNOLOGY STACK

Google Colaboratory

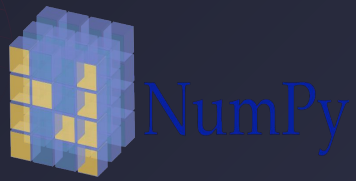
Colaboratory is a free Jupyter notebook environment provided by Google where you can use free GPUs and TPUs.

Tenserflow

It is an open source artificial intelligence library which allows developers to create large-scale neural networks.

Keras

Keras is an Open Source Neural Network library written in Python and a high-level API wrapper.



05

APPLICATIONS

APPLICATIONS

SELF-DRIVING CARS AND ROBOTICS

SECURITY AND VIDEO ANALYSIS

AID TO VISUALLY IMPAIRED

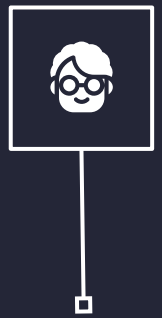
ENCAPSULATING CAPTIONS WITH IMAGE

E-COMMERCE ASSISTANT AND ENHANCING IMAGE SEARCH

References

- Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge --- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan
- Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning --- Piyush Sharma, Nan Ding, Sebastian Goodman, Radu Soricut.
- Image Captioning Based on Deep Neural Networks --- Shuang Liu, Liang Bai, Yanli Hu and Haoran Wang

MEET THE TEAM

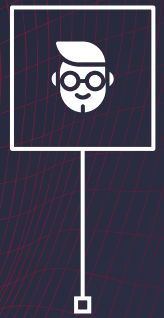


Prof. Sachin Malave

Project Guide

Mrunal S Jadhav

Team Member 1



Aditya G Joshi

Team Member 2

THANK YOU!