DATA SCIENCE BRAIN
@datasciencebrain

COMPUTER VISION CHEAT SHEET

Using OpenCV & Tensorflow

Save for later reference ·················›

01 OPENCV IMAGE LOADING AND DISPLAY

```
import cv2

# Read an image
img = cv2.imread('image.jpg')

# Display the image
cv2.imshow('Image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

02 OPENCV IMAGE OPERATIONS

```
# Convert to grayscale
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Resize an image
resized_img = cv2.resize(img, (width, height))

# Crop an image
cropped_img = img[y1:y2, x1:x2]
```

03 OPENCV IMAGE FILTERING

```
# Gaussian blur
blurred_img = cv2.GaussianBlur(img, (kernel_size, kernel_size), 0)

# Edge detection
edges = cv2.Canny(gray_img, low_threshold, high_threshold)
```

04 OPENCV OBJECT DETECTION

```
# Load Haarcascades classifier
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Detect faces
faces = face_cascade.detectMultiScale(gray_img, scaleFactor=1.3, minNeighbors=5)
```

05 TENSORFLOW IMAGE PREPROCESSING

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

img_path = 'image.jpg'
img = image.load_img(img_path, target_size=(224, 224))
img_array = image.img_to_array(img)
img_array = preprocess_input(img_array)
img_array = np.expand_dims(img_array, axis=0)
```

06 MODEL PREDICTION

```
predictions = model.predict(img_array)
```

07 TRANSFER LEARNING

```
from tensorflow.keras.applications import VGG16
from tensorflow.keras import models, layers

base_model = VGG16(weights='imagenet',
include_top=False, input_shape=(224, 224, 3))

model = models.Sequential()
model.add(base_model)
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(num_classes,
activation='softmax'))
```

08 OBJECT DETECTION

```
# Install the TensorFlow Object Detection API
# Follow the installation guide:
https://github.com/tensorflow/models/blob/master/research/object_detection/
g3doc/tf2.md

from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as vis_util

# Load model and labels
model = tf.saved_model.load('path/to/saved_model')
category_index =
label_map_util.create_category_index_from_labelmap('path/to/label_map.pbtxt',
use_display_name=True)

# Run inference
input_tensor = tf.convert_to_tensor(np.expand_dims(img, 0), dtype=tf.float32)
detections = model(input_tensor)

# Visualize detections
vis_util.visualize_boxes_and_labels_on_image_array(
   img,
   np.squeeze(detections['detection_boxes']),
   np.squeeze(detections['detection_classes']).astype(np.int32),
   np.squeeze(detections['detection_scores']),
   category_index,
   use_normalized_coordinates=True,
   line_thickness=8, )
```

09 IMAGE CLASSIFICATION

```
from tensorflow.keras.applications.inception_v3 import
InceptionV3
from tensorflow.keras.applications.inception_v3 import
preprocess_input, decode_predictions

# Load pre-trained model
model = InceptionV3(weights='imagenet')

# Preprocess and predict
img_array = preprocess_input(img_array)
predictions = model.predict(img_array)

# Decode predictions
decoded_predictions = decode_predictions(predictions,
top=3)[0]
for i, (imagenet_id, label, score) in
enumerate(decoded_predictions):
    print(f"{i + 1}: {label} ({score:.2f})")
```

10 IMAGE SEGMENTATION

```
# Install the TensorFlow Model Garden and the DeepLabV3 model
# Follow the installation guide:
https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/installation.md

from PIL import Image
from matplotlib import pyplot as plt

# Load DeepLabV3 model
model = tf.saved_model.load('path/to/deeplabv3')

# Preprocess image
input_array = tf.image.resize(input_array, (256, 256))
input_array = tf.expand_dims(input_array, 0)

# Run inference
predictions = model(input_array)['segmentation_mask']
predictions = tf.argmax(predictions, axis=-1)

# Visualize segmentation mask
plt.imshow(predictions[0])
plt.show()
```