(/)

Machine Learning Pipeline
Average: **100.0%**

Data Augmentation

100%

Amateur

By: Alexa Orrico, Software Engineer at Holberton School

Weight: 1

Project will start Nov 3, 2024 12:00 AM, must end by Nov 9, 2024 11:59 PM

Resources

Read or watch:

- Data Augmentation | How to use Deep Learning when you have Limited Data—Part 2 (/rltoken/UoIDdYHjbEb8CKIgplxTjw)
- tf.image (/rltoken/LJOfCI3-JLqo8HwMdJqOOg)
- tf.keras.preprocessing.image (/rltoken/5wwkCD_rJh4ttEsqdhAkwQ)
- Automating Data Augmentation: Practice, Theory and New Direction (/rltoken/IAePg_uphRa9guBabgIoJA)

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/1jKnHaOpiDnvkBk_GCFvyA), **without the help of Google**:

General

- What is data augmentation?
- When should you perform data augmentation?
- What are the benefits of using data augmentation?
- What are the various ways to perform data augmentation?
- How can you use ML to automate data augmentation?

Requirements

General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted/compiled on Ubuntu 16.04 LTS using `python3` (version 3.6.12)
- Your files will be executed with `numpy` (version 1.16) and `tensorflow` (version 1.15)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/env python3`
- All of your files must be executable
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should follow the `pycodestyle` style (version 2.4)
- All your modules should have documentation (`python3 -c 'print(__import__("my_module").__doc__)'`)
- All your classes should have documentation (`python3 -c 'print(__import__("my_module").MyClass.__doc__)'`)
- All your functions (inside and outside a class) should have documentation (`python3 -c 'print(__import__("my_module").my_function.__doc__)'` and `python3 -c 'print(__import__("my_module").MyClass.my_function.__doc__)'`)
- Unless otherwise stated, you cannot import any module except `import tensorflow as tf`

Download TF Datasets

```
pip install --user tensorflow-datasets
```

Tasks

0. Flip

mandatory

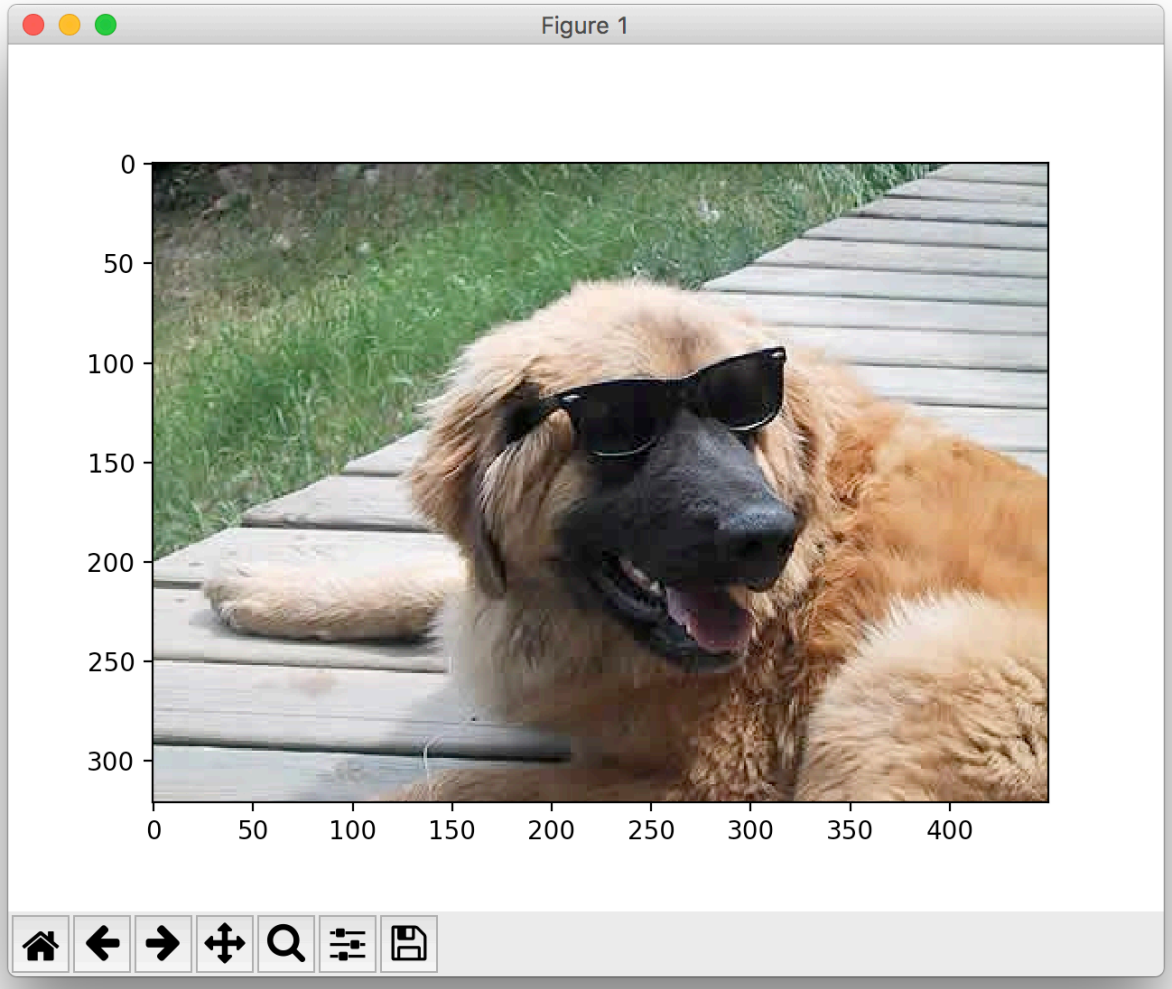
Write a function `def flip_image(image):` that flips an image horizontally:

- `image` is a 3D `tf.Tensor` containing the image to flip
- Returns the flipped image

```
$ cat 0-main.py
#!/usr/bin/env python3
"""
import tensorflow as tf
import tensorflow_datasets as tfds
import matplotlib.pyplot as plt
flip_image = __import__('0-flip').flip_image

tf.compat.v1.enable_eager_execution()
tf.compat.v1.set_random_seed(0)

doggies = tfds.load('stanford_dogs', split='train', as_supervised=True)
for image, _ in doggies.shuffle(10).take(1):
    plt.imshow(flip_image(image))
    plt.show()
$ ./0-main.py
```



Repo:

- GitHub repository: `alu-machine_learning`
- Directory: `pipeline/data_augmentation`
- File: `0-flip.py`

Please review your task manually with the following checklist

✓

Output is the same as the main file example

5/5 pts

1. Crop

mandatory

Write a function `def crop_image(image, size):` that performs a random crop of an image:

- `image` is a 3D `tf.Tensor` containing the image to crop
- `size` is a tuple containing the size of the crop
- Returns the cropped image

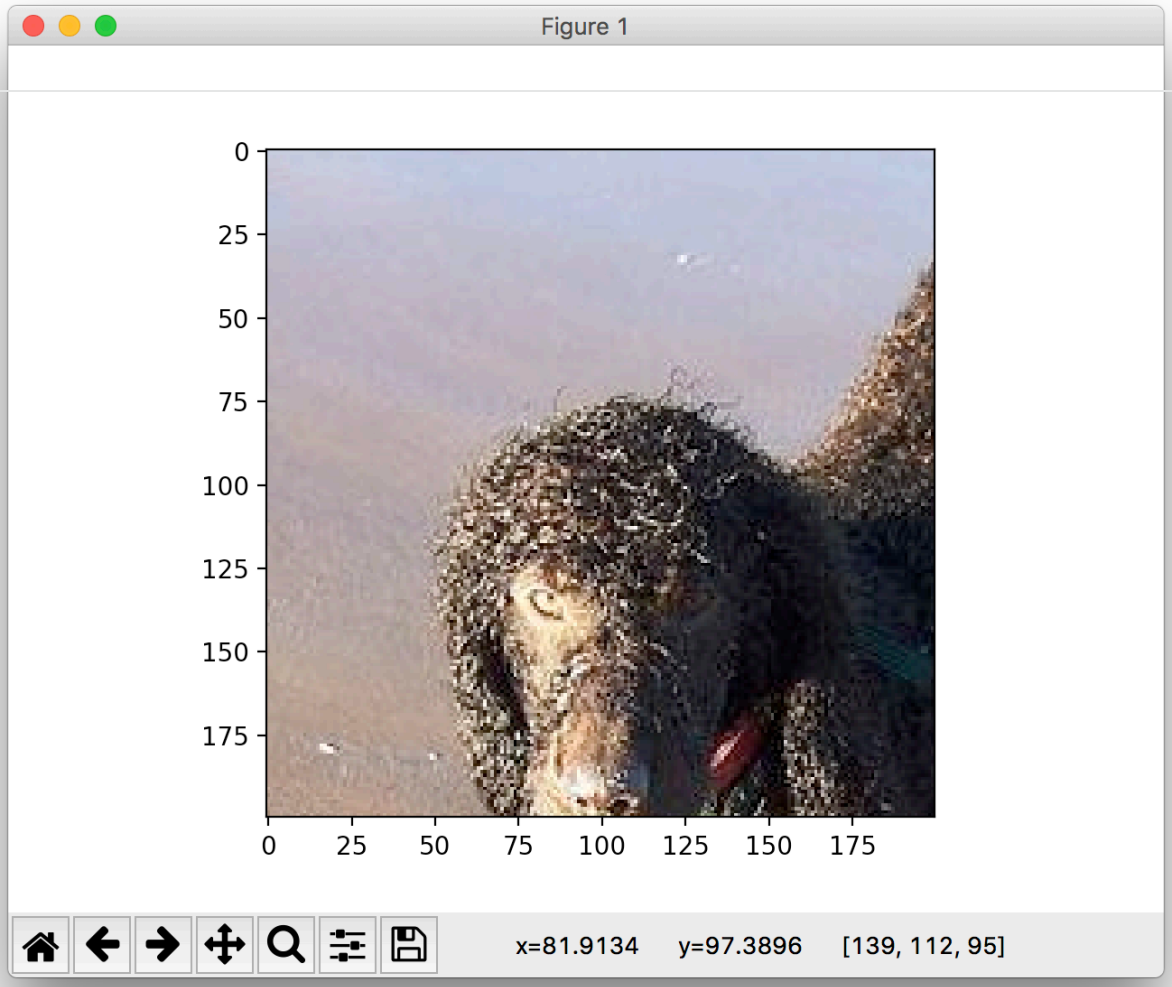
```
$ cat 1-main.py
#!/usr/bin/env python3

import tensorflow as tf
import tensorflow_datasets as tfds
import matplotlib.pyplot as plt
crop_image = __import__('1-crop').crop_image

tf.compat.v1.enable_eager_execution()
tf.compat.v1.set_random_seed(1)

doggies = tfds.load('stanford_dogs', split='train', as_supervised=True)
for image, _ in doggies.shuffle(10).take(1):
    plt.imshow(crop_image(image, (200, 200, 3)))
    plt.show()
$ ./1-main.py
```

(/)



Repo:

- GitHub repository: `alu-machine_learning`
- Directory: `pipeline/data_augmentation`
- File: `1-crop.py`

Please review your task manually with the following checklist

✓

Output is the same as the main file

5/5 pts

2. Rotate

mandatory

(/)

Write a function `def rotate_image(image):` that rotates an image by 90 degrees counter-clockwise:

- `image` is a 3D `tf.Tensor` containing the image to rotate
- Returns the rotated image

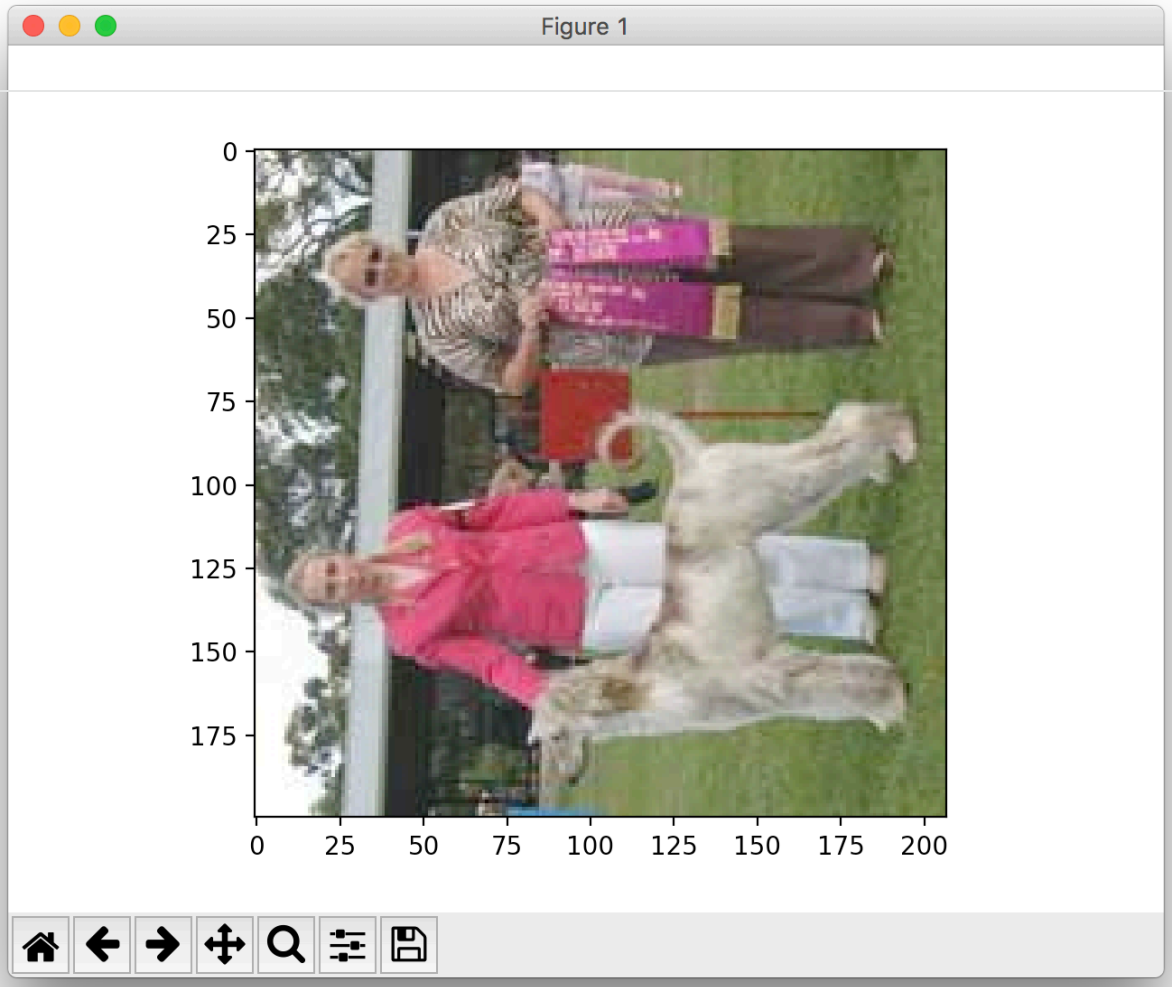
```
$ cat 2-main.py
#!/usr/bin/env python3

import tensorflow as tf
import tensorflow_datasets as tfds
import matplotlib.pyplot as plt
rotate_image = __import__('2-rotate').rotate_image

tf.compat.v1.enable_eager_execution()
tf.compat.v1.set_random_seed(2)

doggies = tfds.load('stanford_dogs', split='train', as_supervised=True)
for image, _ in doggies.shuffle(10).take(1):
    plt.imshow(rotate_image(image))
    plt.show()
$ ./2-main.py
```

(/)



Repo:

- GitHub repository: `alu-machine_learning`
- Directory: `pipeline/data_augmentation`
- File: `2-rotate.py`

Please review your task manually with the following checklist

✓

Output is the same as the main file

5/5 pts

3. Shear

mandatory

(/)

Write a function `def shear_image(image, intensity):` that randomly shears an image:

- `image` is a 3D `tf.Tensor` containing the image to shear
- `intensity` is the intensity with which the image should be sheared
- Returns the sheared image

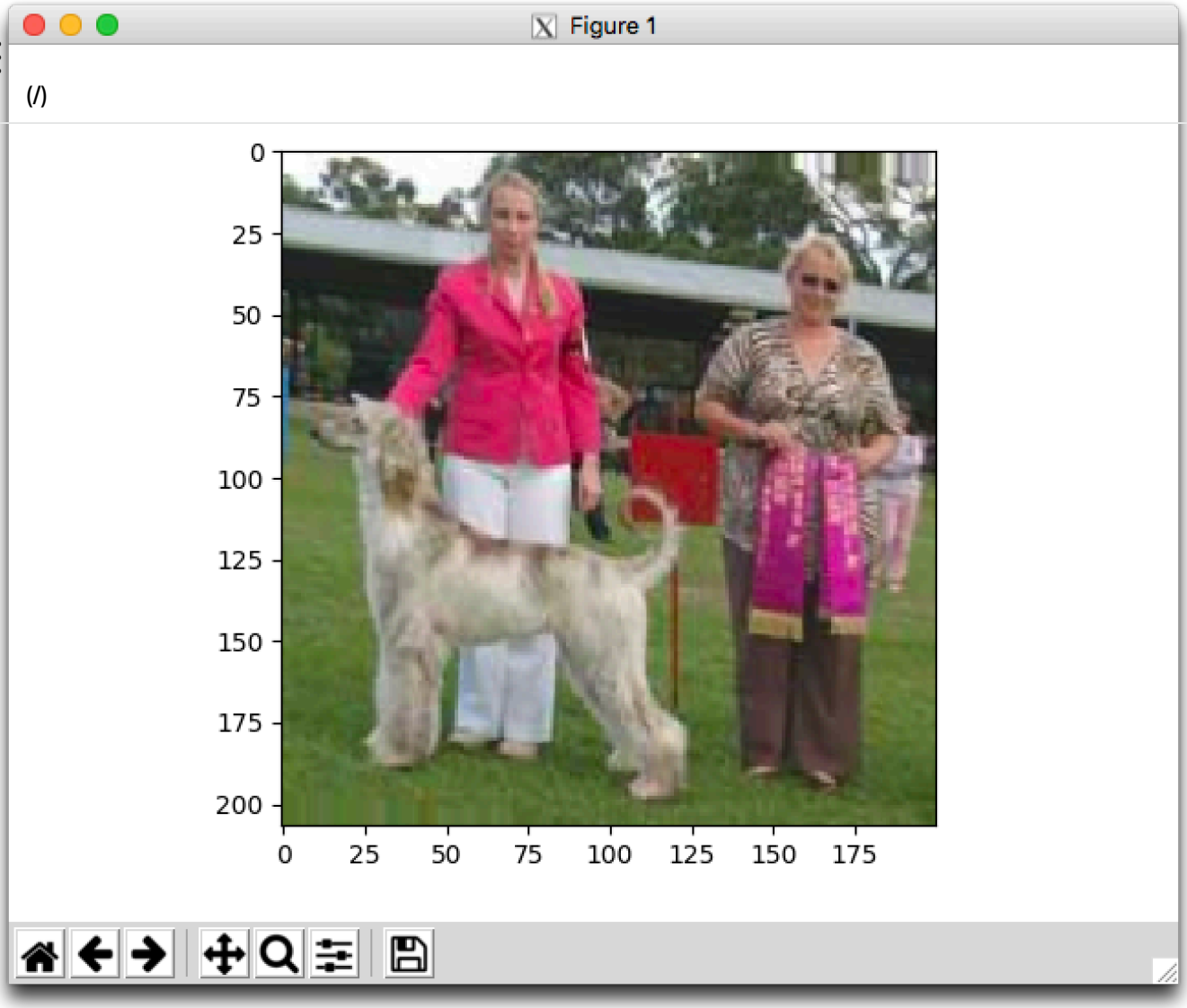
```
$ cat 3-main.py
#!/usr/bin/env python3

import tensorflow as tf
import tensorflow_datasets as tfds
import matplotlib.pyplot as plt
shear_image = __import__('3-shear').shear_image

tf.compat.v1.enable_eager_execution()
tf.compat.v1.set_random_seed(3)

doggies = tfds.load('stanford_dogs', split='train', as_supervised=True)
for image, _ in doggies.shuffle(10).take(1):
    plt.imshow(shear_image(image, 50))
    plt.show()
$ ./3-main.py
```

(/)



Repo:

- GitHub repository: `alu-machine_learning`
- Directory: `pipeline/data_augmentation`
- File: `3-shear.py`

Please review your task manually with the following checklist

✓

Output is the same as the main file

5/5 pts

4. Brightness

mandatory

(/)

Write a function `def change_brightness(image, max_delta):` that randomly changes the brightness of an image:

- `image` is a 3D `tf.Tensor` containing the image to change
- `max_delta` is the maximum amount the image should be brightened (or darkened)
- Returns the altered image

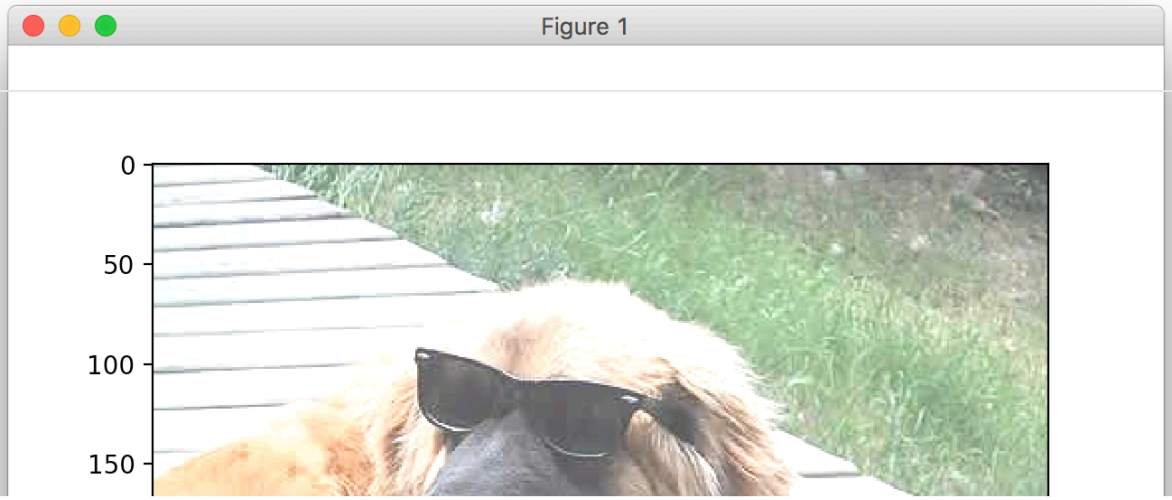
```
$ cat 4-main.py
#!/usr/bin/env python3

import tensorflow as tf
import tensorflow_datasets as tfds
import matplotlib.pyplot as plt
change_brightness = __import__('4-brightness').change_brightness

tf.compat.v1.enable_eager_execution()
tf.compat.v1.set_random_seed(4)

doggies = tfds.load('stanford_dogs', split='train', as_supervised=True)
for image, _ in doggies.shuffle(10).take(1):
    plt.imshow(change_brightness(image, 0.3))
    plt.show()
$ ./4-main.py
```

(/)



Repo:

- GitHub repository: `alu-machine_learning`
- Directory: `pipeline/data_augmentation`
- File: `4-brightness.py`

Please review your task manually with the following checklist

✓

Output is the same as the main file

5/5 pts

5. Hue

mandatory

(/)

Write a function `def change_hue(image, delta):` that changes the hue of an image:

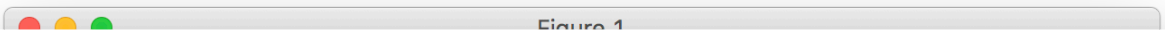
- `image` is a 3D `tf.Tensor` containing the image to change
- `delta` is the amount the hue should change
- Returns the altered image

```
$ cat 5-main.py
#!/usr/bin/env python3

import tensorflow as tf
import tensorflow_datasets as tfds
import matplotlib.pyplot as plt
change_hue = __import__('5-hue').change_hue

tf.compat.v1.enable_eager_execution()
tf.compat.v1.set_random_seed(5)

doggies = tfds.load('stanford_dogs', split='train', as_supervised=True)
for image, _ in doggies.shuffle(10).take(1):
    plt.imshow(change_hue(image, -0.5))
    plt.show()
$ ./5-main.py
```



(/)

Repo:

- GitHub repository: `alu-machine_learning`
- Directory: `pipeline/data_augmentation`
- File: `5-hue.py`

Please review your task manually with the following checklist

✓

Output is the same as the main file

5/5 pts

6. Automation

mandatory

(/)

Write a blog post describing step by step how to perform automated data augmentation. Try to explain every step you know of, and give examples. A total beginner should understand what you have written.

- Have at least one picture, at the top of the blog post
- Publish your blog post on Medium or LinkedIn
- Share your blog post at least on LinkedIn
- Write professionally and intelligibly
- Please, remember that these blogs must be written in English to further your technical ability in a variety of settings

Remember, future employers will see your articles; take this seriously, and produce something that will be an asset to your future

When done, please add all urls below (blog post, LinkedIn post, etc.)

Add URLs here:

Save

1. https://medium.com/@m.yasin/data-augmentation-simple-guide-3d3bd131da70 (https://medium.com/@m.yasin/data-augmentation-simple-guide-3d3bd131da70) Remove

Please review your task manually with the following checklist

TODO

5/5 pts

Done with the mandatory tasks? Unlock 1 advanced task now!

Score

(/)

100%

Congratulations! You made it!

The next project will be available on Sunday, Nov 10th.

Go to home (/)

Previous project (/projects/2420)