MAX POOLING

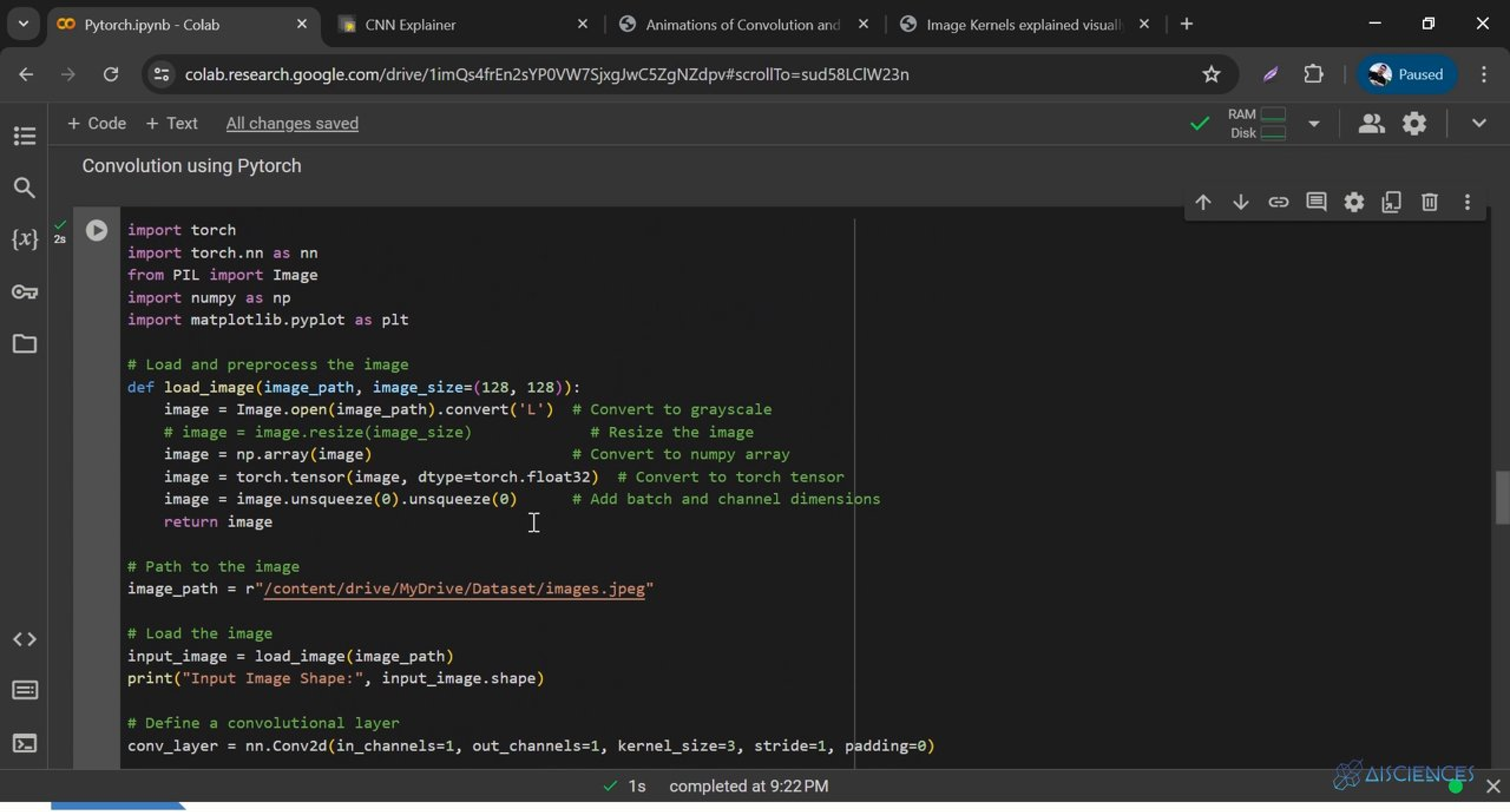
Input

$$\begin{bmatrix}6 & 3 & 2\\ 5 & 2 & 1\\ 4 & 1 & 0\end{bmatrix}$$

Pooling matrix
2x2

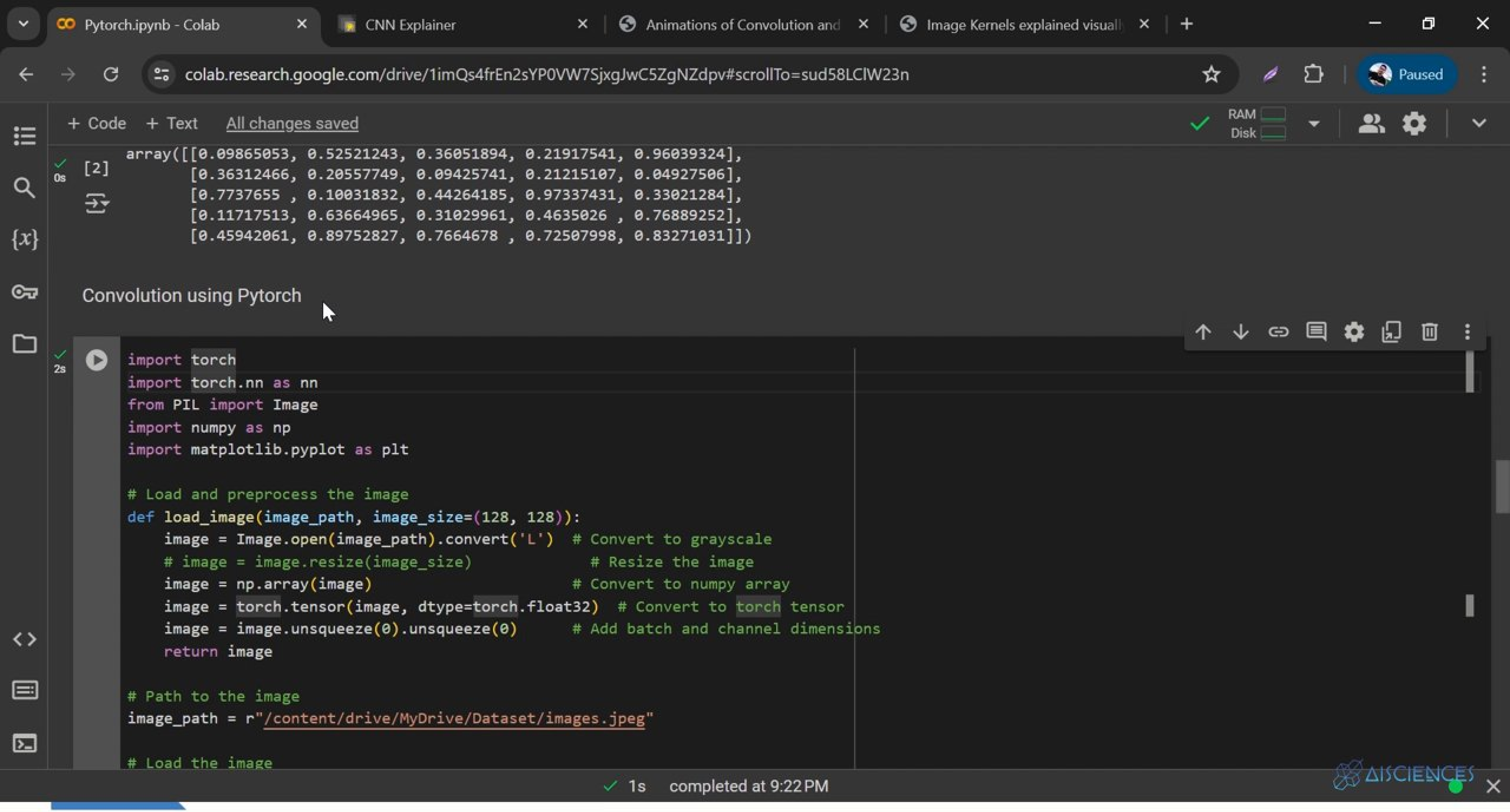
Output

$$\begin{bmatrix}6 & 2\\ 4 & 0\end{bmatrix}$$


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
All changes saved
RAM
Disk
2s
1s
completed at 9:22 PM
AISCIENCES

Convolution using Pytorch

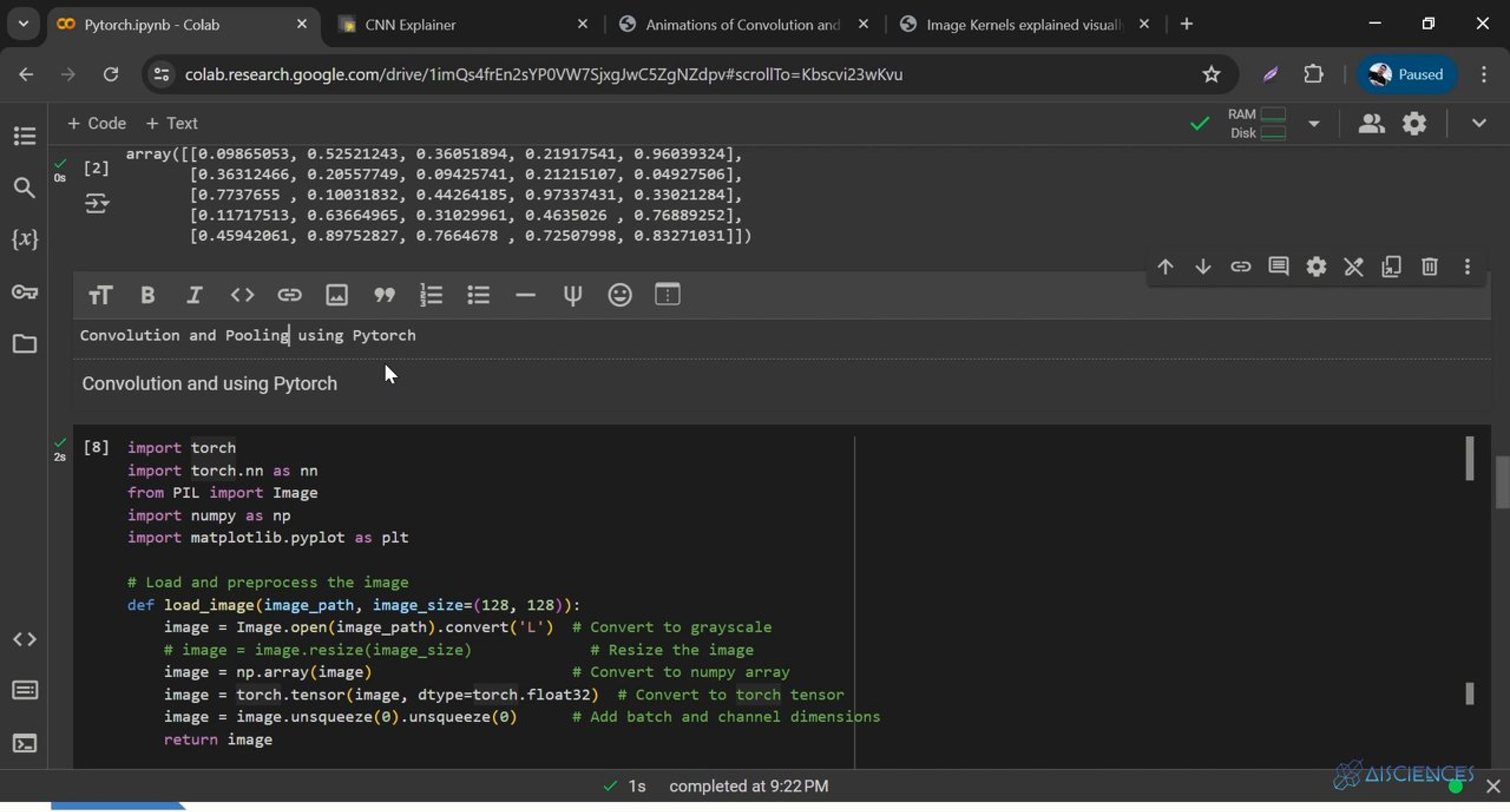
```python
import torch
import torch.nn as nn
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt

# Load and preprocess the image
def load_image(image_path, image_size=(128, 128)):
    image = Image.open(image_path).convert('L')  # Convert to grayscale
    # image = image.resize(image_size)            # Resize the image
    image = np.array(image)                      # Convert to numpy array
    image = torch.tensor(image, dtype=torch.float32)  # Convert to torch tensor
    image = image.unsqueeze(0).unsqueeze(0)      # Add batch and channel dimensions
    return image

# Path to the image
image_path = r"/content/drive/MyDrive/Dataset/images.jpeg"

# Load the image
input_image = load_image(image_path)
print("Input Image Shape:", input_image.shape)

# Define a convolutional layer
conv_layer = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=3, stride=1, padding=0)
```


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
All changes saved
RAM
Disk
0s
[2]
2s
✓ 1s completed at 9:22 PM
AISCIENCES

```
array([[0.09865053, 0.52521243, 0.36051894, 0.21917541, 0.96039324],
       [0.36312466, 0.20557749, 0.09425741, 0.21215107, 0.04927506],
       [0.7737655 , 0.10031832, 0.44264185, 0.97337431, 0.33021284],
       [0.11717513, 0.63664965, 0.31029961, 0.4635026 , 0.76889252],
       [0.45942061, 0.89752827, 0.7664678 , 0.72507998, 0.83271031]])
```

Convolution using Pytorch

```python
import torch
import torch.nn as nn
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt

# Load and preprocess the image
def load_image(image_path, image_size=(128, 128)):
    image = Image.open(image_path).convert('L')  # Convert to grayscale
    # image = image.resize(image_size)             # Resize the image
    image = np.array(image)                      # Convert to numpy array
    image = torch.tensor(image, dtype=torch.float32)  # Convert to torch tensor
    image = image.unsqueeze(0).unsqueeze(0)      # Add batch and channel dimensions
    return image

# Path to the image
image_path = r"/content/drive/MyDrive/Dataset/images.jpeg"

# Load the image
```

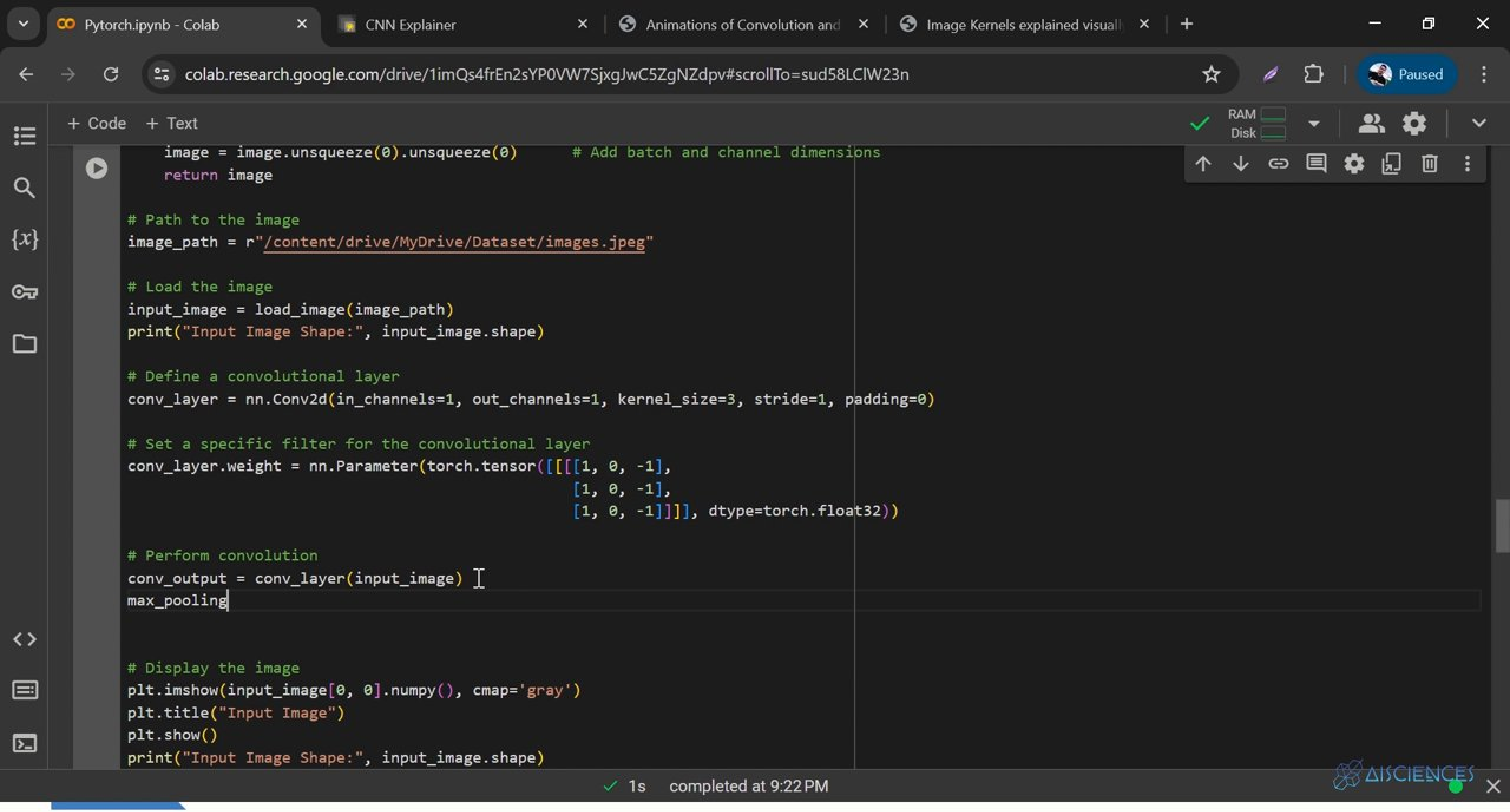

Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=Kbscvi23wKvu
Paused
+ Code
+ Text
RAM
Disk
[2]
0s
[8]
2s
1s completed at 9:22 PM
AISCIENCES

```
array([[0.09865053, 0.52521243, 0.36051894, 0.21917541, 0.96039324],
       [0.36312466, 0.20557749, 0.09425741, 0.21215107, 0.04927506],
       [0.7737655 , 0.10031832, 0.44264185, 0.97337431, 0.33021284],
       [0.11717513, 0.63664965, 0.31029961, 0.4635026 , 0.76889252],
       [0.45942061, 0.89752827, 0.7664678 , 0.72507998, 0.83271031]])
```

```
Convolution and Pooling using Pytorch
```

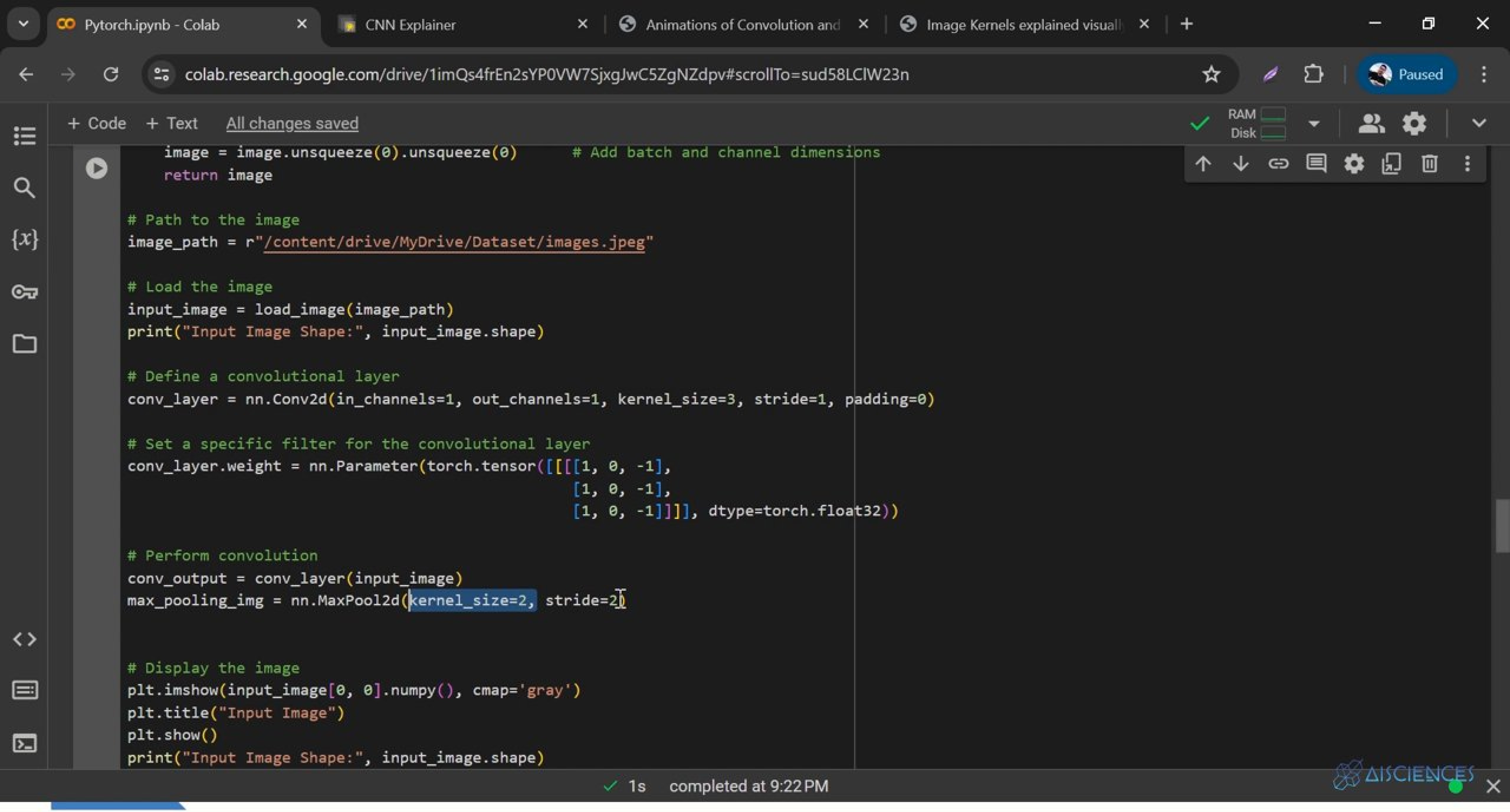
Convolution and using Pytorch

```python
import torch
import torch.nn as nn
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt

# Load and preprocess the image
def load_image(image_path, image_size=(128, 128)):
    image = Image.open(image_path).convert('L')  # Convert to grayscale
    # image = image.resize(image_size)            # Resize the image
    image = np.array(image)                      # Convert to numpy array
    image = torch.tensor(image, dtype=torch.float32)  # Convert to torch tensor
    image = image.unsqueeze(0).unsqueeze(0)      # Add batch and channel dimensions
    return image
```

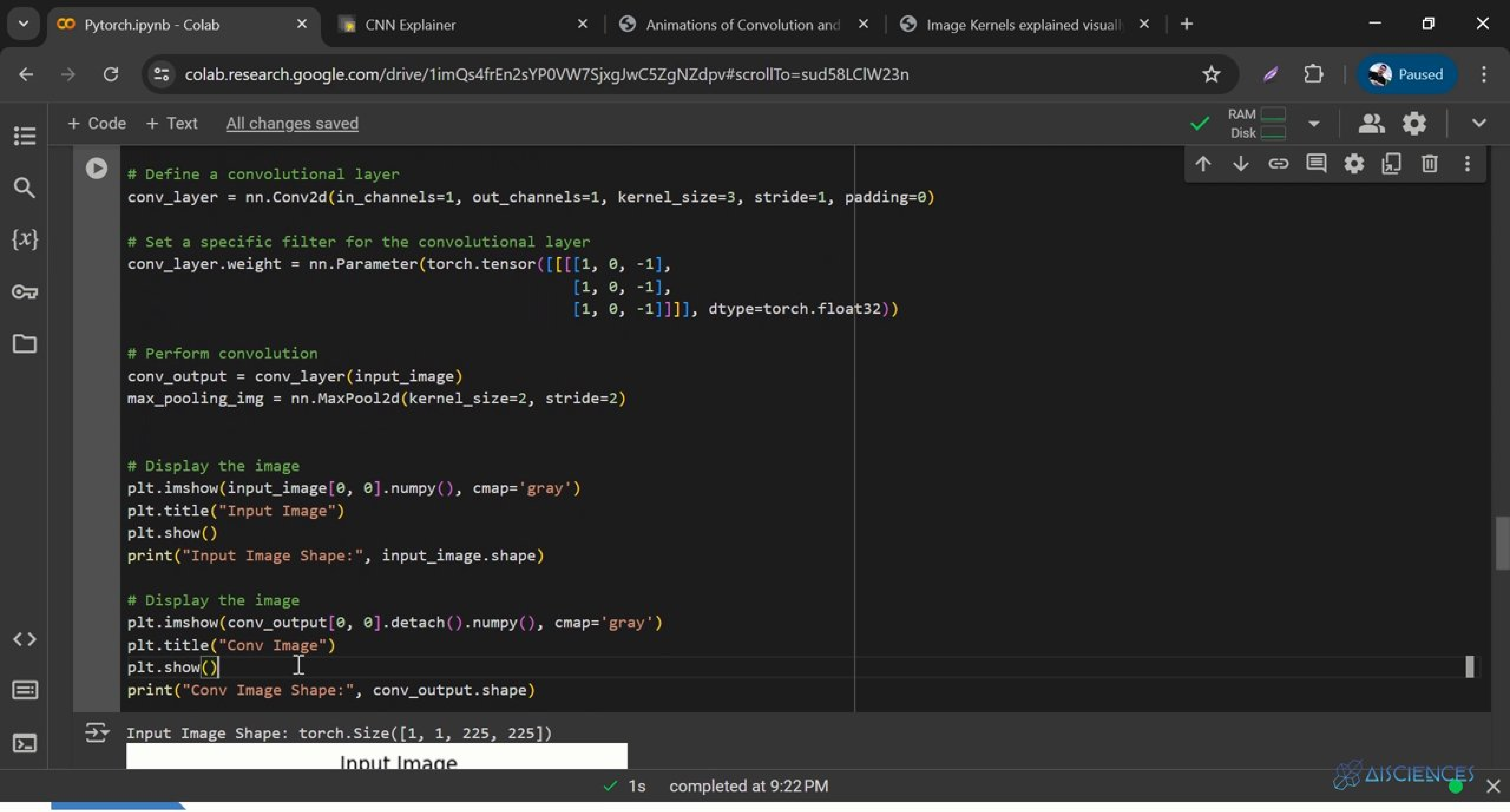


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
RAM
Disk
✓ 1s completed at 9:22 PM
AISCIENCES

```
    image = image.unsqueeze(0).unsqueeze(0)      # Add batch and channel dimensions
    return image

# Path to the image
image_path = r"/content/drive/MyDrive/Dataset/images.jpeg"

# Load the image
input_image = load_image(image_path)
print("Input Image Shape:", input_image.shape)

# Define a convolutional layer
conv_layer = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=3, stride=1, padding=0)

# Set a specific filter for the convolutional layer
conv_layer.weight = nn.Parameter(torch.tensor([[[[1, 0, -1],
                                                 [1, 0, -1],
                                                 [1, 0, -1]]]], dtype=torch.float32))

# Perform convolution
conv_output = conv_layer(input_image)
max_pooling


# Display the image
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)
```


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
All changes saved
RAM
Disk
1s
completed at 9:22 PM
AISCIENCES

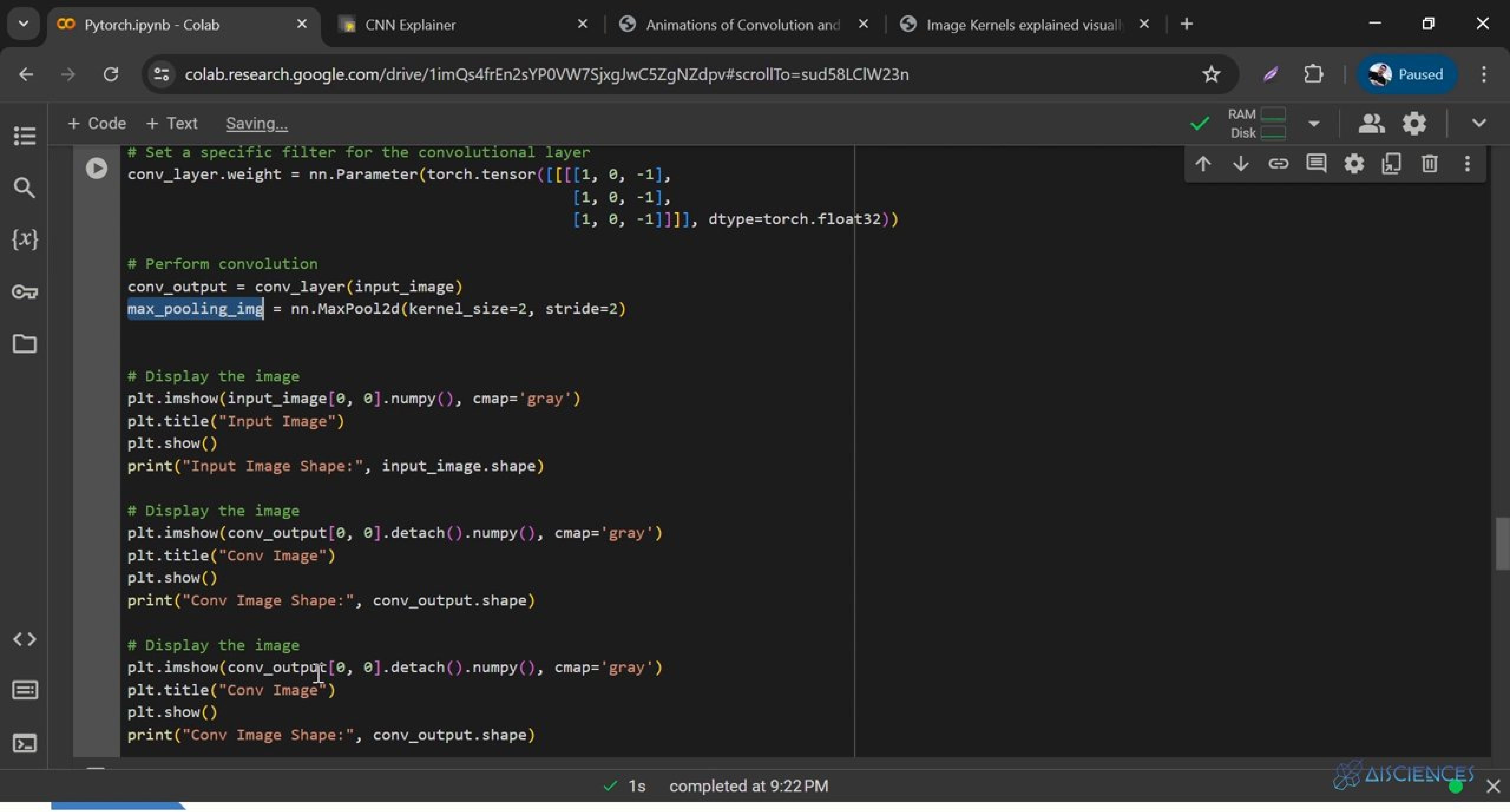
```
    image = image.unsqueeze(0).unsqueeze(0)      # Add batch and channel dimensions
    return image

# Path to the image
image_path = r"/content/drive/MyDrive/Dataset/images.jpeg"

# Load the image
input_image = load_image(image_path)
print("Input Image Shape:", input_image.shape)

# Define a convolutional layer
conv_layer = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=3, stride=1, padding=0)

# Set a specific filter for the convolutional layer
conv_layer.weight = nn.Parameter(torch.tensor([[[[1, 0, -1],
                                                 [1, 0, -1],
                                                 [1, 0, -1]]]], dtype=torch.float32))

# Perform convolution
conv_output = conv_layer(input_image)
max_pooling_img = nn.MaxPool2d(kernel_size=2, stride=2)


# Display the image
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)
```


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
All changes saved
RAM
Disk
1s
completed at 9:22 PM
AISCIENCES

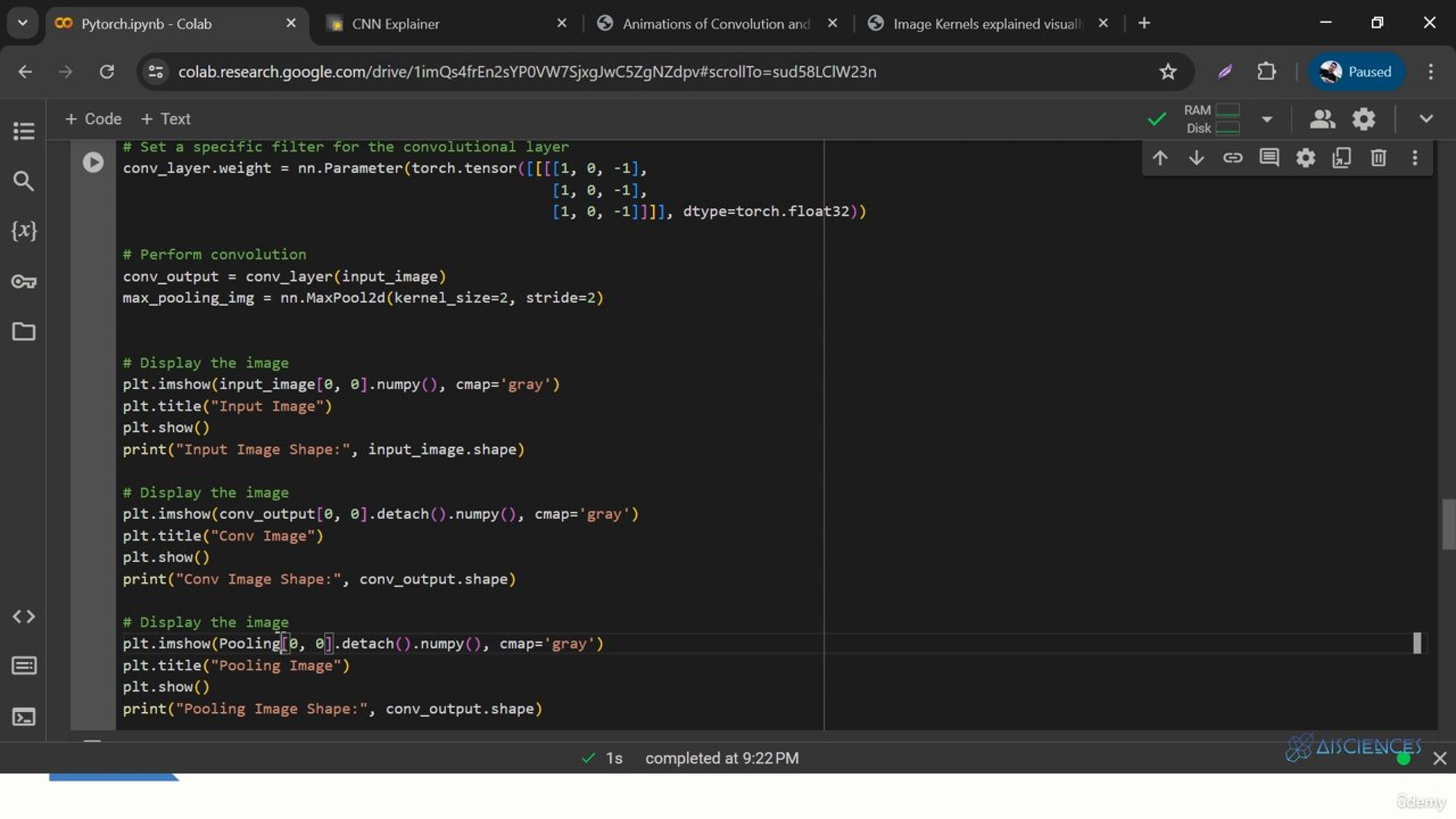
```
# Define a convolutional layer
conv_layer = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=3, stride=1, padding=0)

# Set a specific filter for the convolutional layer
conv_layer.weight = nn.Parameter(torch.tensor([[[[1, 0, -1],
                                                [1, 0, -1],
                                                [1, 0, -1]]]], dtype=torch.float32))

# Perform convolution
conv_output = conv_layer(input_image)
max_pooling_img = nn.MaxPool2d(kernel_size=2, stride=2)


# Display the image
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)

# Display the image
plt.imshow(conv_output[0, 0].detach().numpy(), cmap='gray')
plt.title("Conv Image")
plt.show()
print("Conv Image Shape:", conv_output.shape)
```

```
Input Image Shape: torch.Size([1, 1, 225, 225])
```

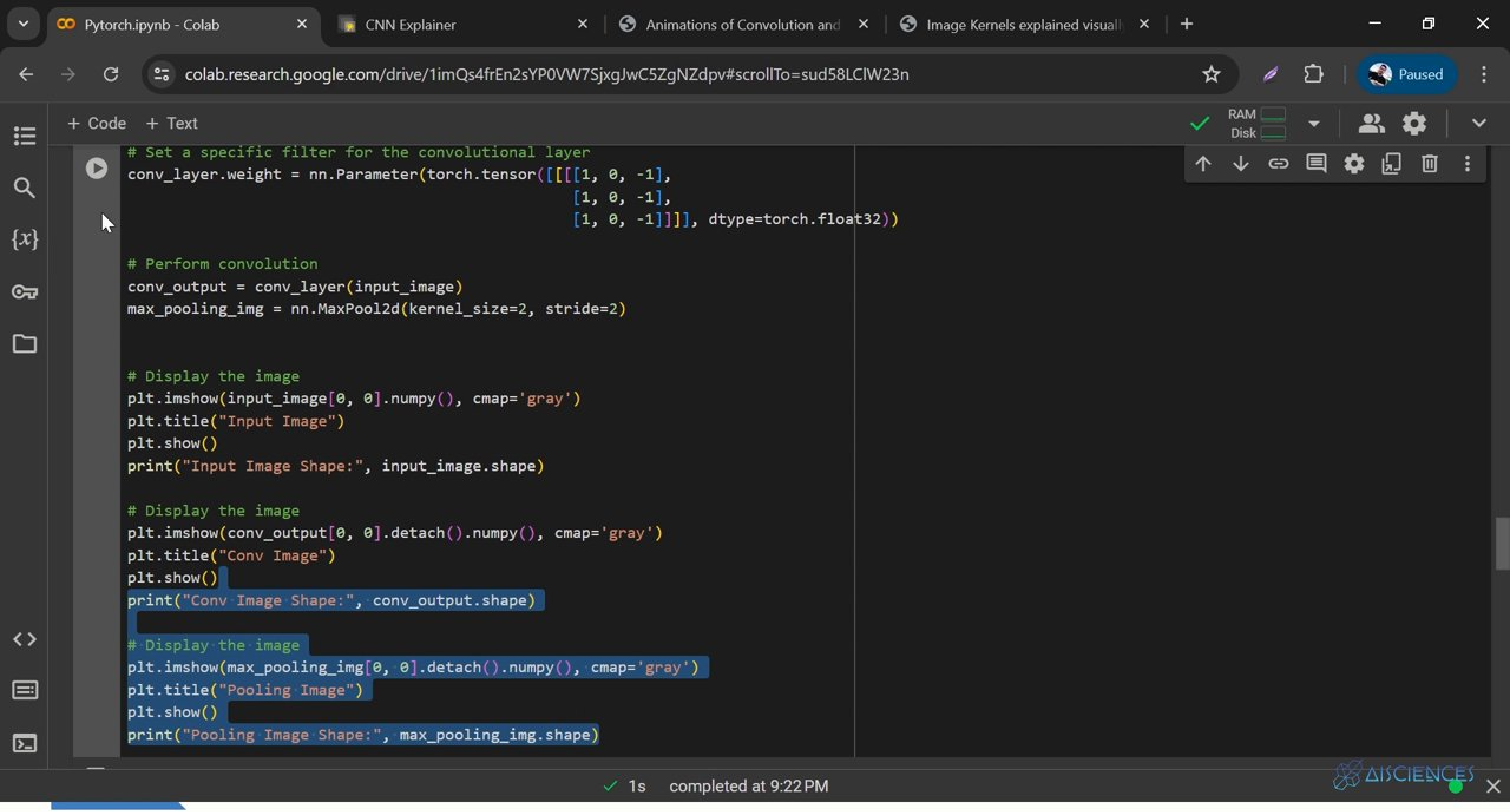
Input Image


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
Saving...
RAM
Disk
1s
completed at 9:22 PM
AISCIENCES

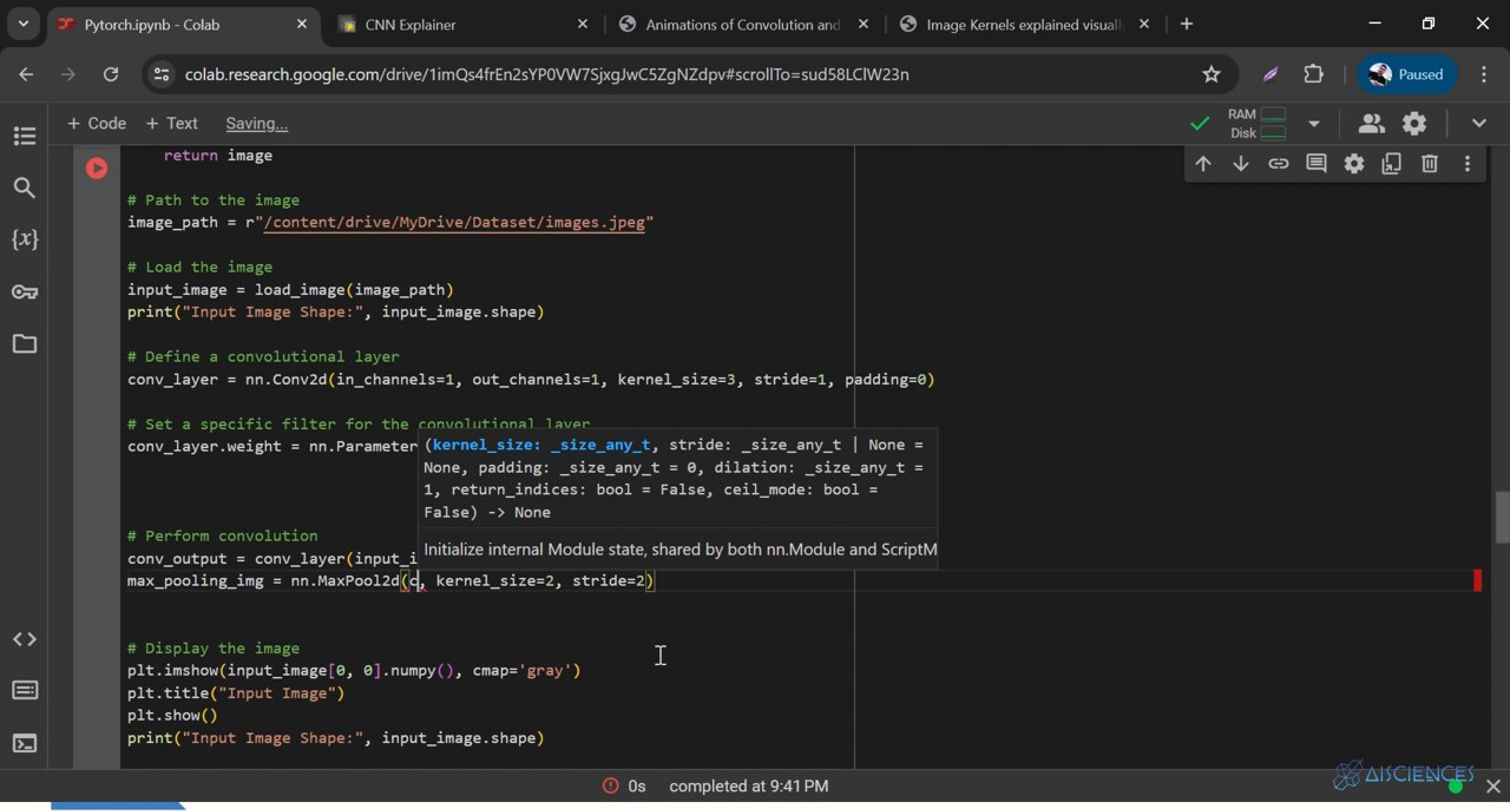
```
# Set a specific filter for the convolutional layer
conv_layer.weight = nn.Parameter(torch.tensor([[[[1, 0, -1],
                                                  [1, 0, -1],
                                                  [1, 0, -1]]]], dtype=torch.float32))

# Perform convolution
conv_output = conv_layer(input_image)
max_pooling_img = nn.MaxPool2d(kernel_size=2, stride=2)


# Display the image
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)

# Display the image
plt.imshow(conv_output[0, 0].detach().numpy(), cmap='gray')
plt.title("Conv Image")
plt.show()
print("Conv Image Shape:", conv_output.shape)

# Display the image
plt.imshow(conv_output[0, 0].detach().numpy(), cmap='gray')
plt.title("Conv Image")
plt.show()
print("Conv Image Shape:", conv_output.shape)
```


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
RAM
Disk
1s
completed at 9:22 PM
AISCIENCES
Udemy

```
# Set a specific filter for the convolutional layer
conv_layer.weight = nn.Parameter(torch.tensor([[[[1, 0, -1],
                                                 [1, 0, -1],
                                                 [1, 0, -1]]]], dtype=torch.float32))

# Perform convolution
conv_output = conv_layer(input_image)
max_pooling_img = nn.MaxPool2d(kernel_size=2, stride=2)


# Display the image
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)

# Display the image
plt.imshow(conv_output[0, 0].detach().numpy(), cmap='gray')
plt.title("Conv Image")
plt.show()
print("Conv Image Shape:", conv_output.shape)

# Display the image
plt.imshow(Pooling[0, 0].detach().numpy(), cmap='gray')
plt.title("Pooling Image")
plt.show()
print("Pooling Image Shape:", conv_output.shape)
```


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LCIW23n
Paused
+ Code
+ Text
RAM
Disk
✓ 1s completed at 9:22 PM
AISCIENCES

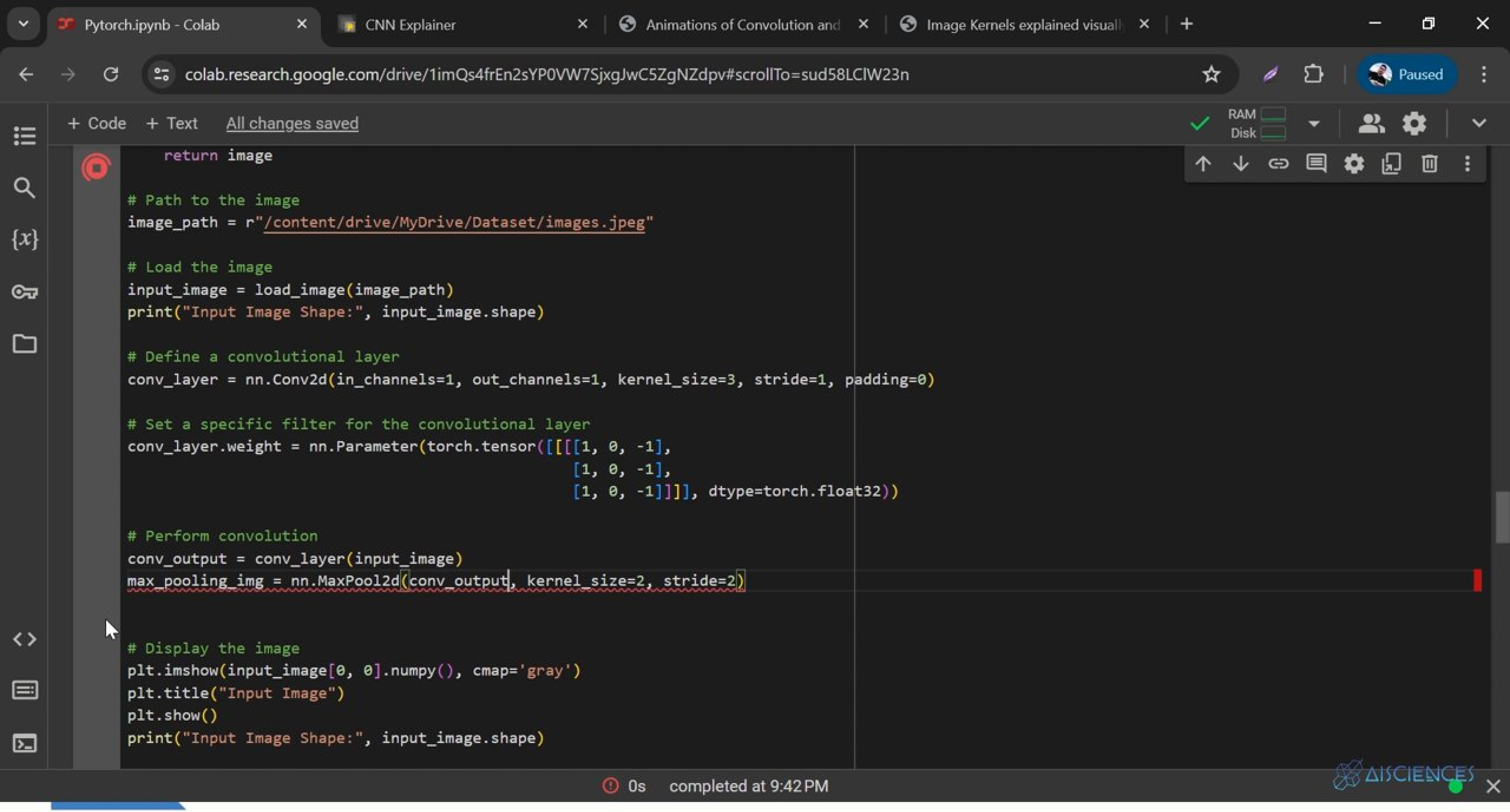
```
# Set a specific filter for the convolutional layer
conv_layer.weight = nn.Parameter(torch.tensor([[[[1, 0, -1],
                                                 [1, 0, -1],
                                                 [1, 0, -1]]]], dtype=torch.float32))


# Perform convolution
conv_output = conv_layer(input_image)
max_pooling_img = nn.MaxPool2d(kernel_size=2, stride=2)



# Display the image
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)

# Display the image
plt.imshow(conv_output[0, 0].detach().numpy(), cmap='gray')
plt.title("Conv Image")
plt.show()
print("Conv Image Shape:", conv_output.shape)

# Display the image
plt.imshow(max_pooling_img[0, 0].detach().numpy(), cmap='gray')
plt.title("Pooling Image")
plt.show()
print("Pooling Image Shape:", max_pooling_img.shape)
```


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
Saving...
RAM
Disk
AISCIENCES

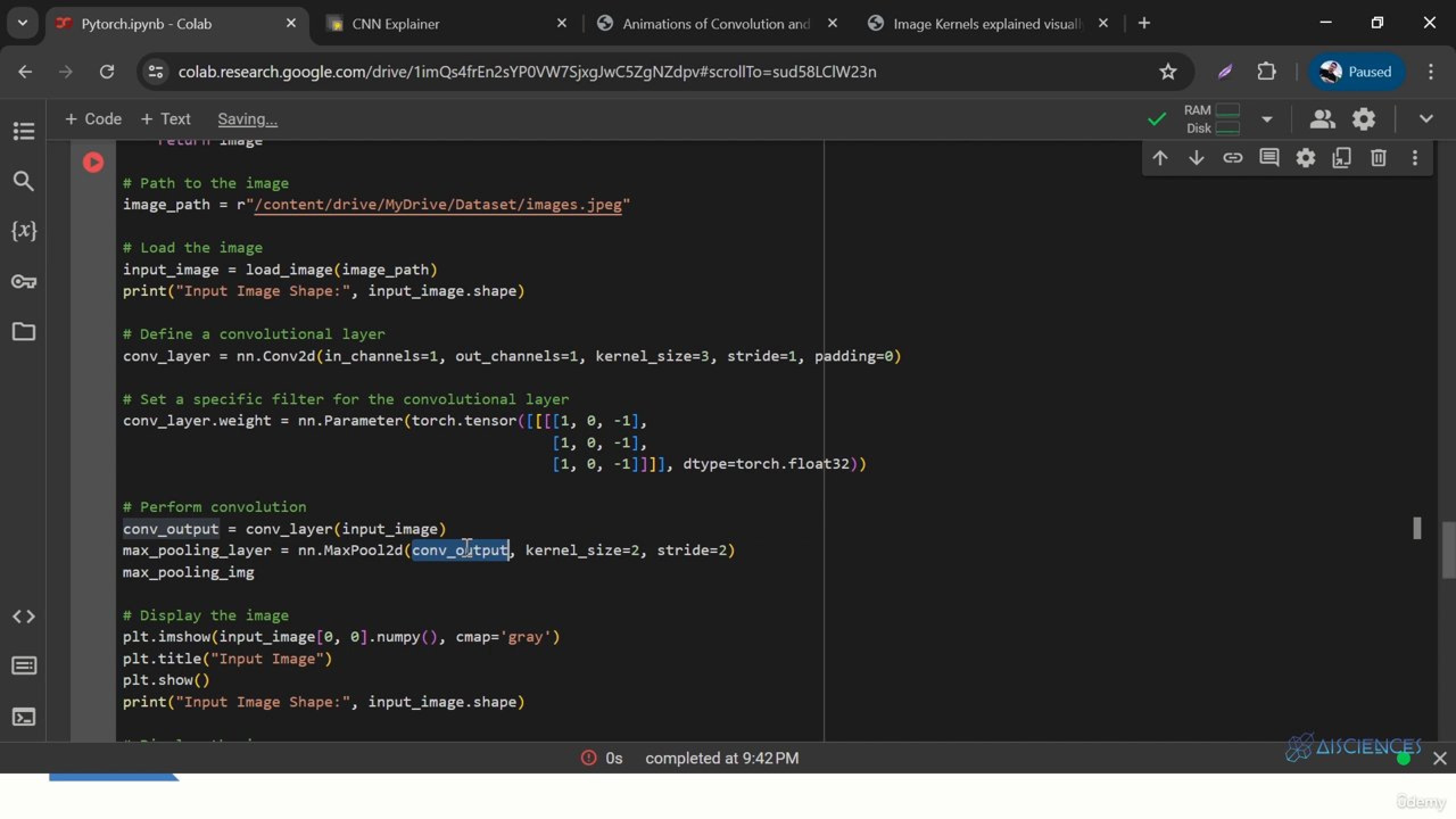
```
    return image

# Path to the image
image_path = r"/content/drive/MyDrive/Dataset/images.jpeg"

# Load the image
input_image = load_image(image_path)
print("Input Image Shape:", input_image.shape)

# Define a convolutional layer
conv_layer = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=3, stride=1, padding=0)

# Set a specific filter for the convolutional layer
conv_layer.weight = nn.Parameter



# Perform convolution
conv_output = conv_layer(input_i
max_pooling_img = nn.MaxPool2d(c, kernel_size=2, stride=2)


# Display the image
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)
```

```
(kernel_size: _size_any_t, stride: _size_any_t | None =
None, padding: _size_any_t = 0, dilation: _size_any_t =
1, return_indices: bool = False, ceil_mode: bool =
False) -> None

Initialize internal Module state, shared by both nn.Module and ScriptM
```

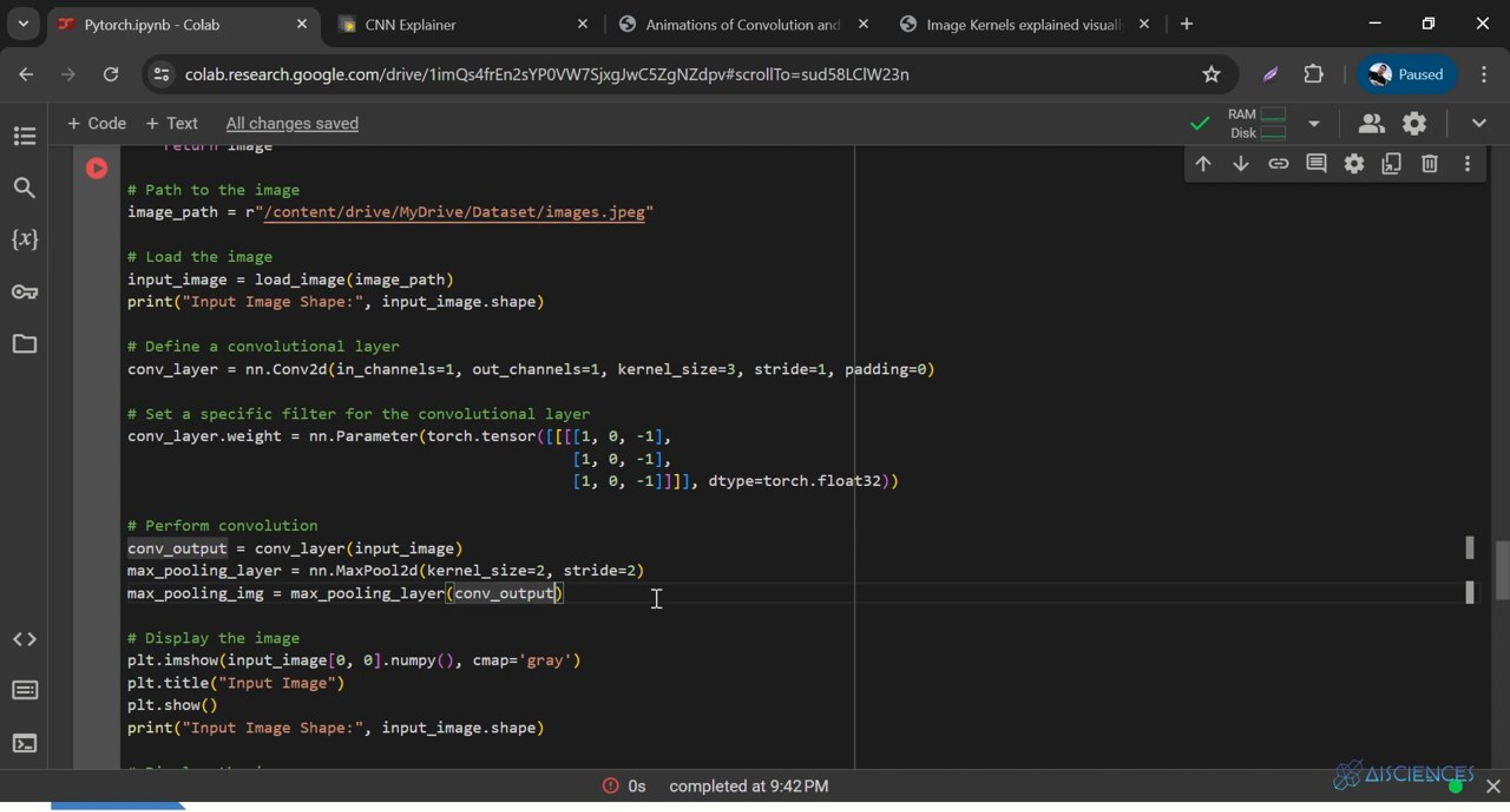
0s completed at 9:41 PM


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
All changes saved
RAM
Disk
0s
completed at 9:42 PM

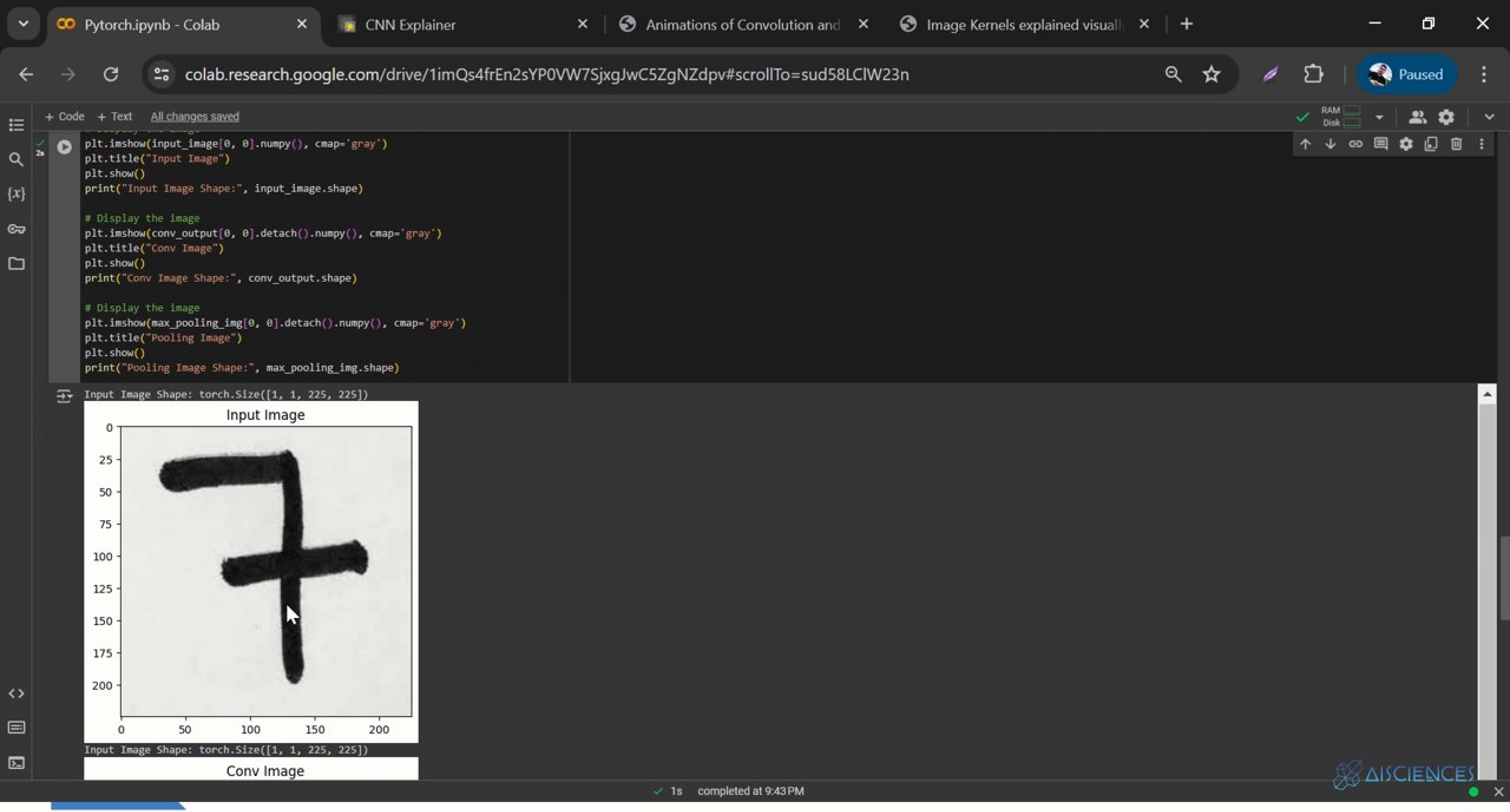
```
    return image

# Path to the image
image_path = r"/content/drive/MyDrive/Dataset/images.jpeg"

# Load the image
input_image = load_image(image_path)
print("Input Image Shape:", input_image.shape)

# Define a convolutional layer
conv_layer = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=3, stride=1, padding=0)

# Set a specific filter for the convolutional layer
conv_layer.weight = nn.Parameter(torch.tensor([[[[1, 0, -1],
                                                 [1, 0, -1],
                                                 [1, 0, -1]]]], dtype=torch.float32))

# Perform convolution
conv_output = conv_layer(input_image)
max_pooling_img = nn.MaxPool2d(conv_output, kernel_size=2, stride=2)


# Display the image
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)
```


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
Saving...
RAM
Disk
0s
completed at 9:42 PM
AISCIENCES

```
    return image

# Path to the image
image_path = r"/content/drive/MyDrive/Dataset/images.jpeg"

# Load the image
input_image = load_image(image_path)
print("Input Image Shape:", input_image.shape)

# Define a convolutional layer
conv_layer = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=3, stride=1, padding=0)

# Set a specific filter for the convolutional layer
conv_layer.weight = nn.Parameter(torch.tensor([[[[1, 0, -1],
                                                  [1, 0, -1],
                                                  [1, 0, -1]]]], dtype=torch.float32))

# Perform convolution
conv_output = conv_layer(input_image)
max_pooling_layer = nn.MaxPool2d(conv_output, kernel_size=2, stride=2)
max_pooling_img

# Display the image
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)
```


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
All changes saved
RAM
Disk
0s
completed at 9:42 PM
AISCIENCES

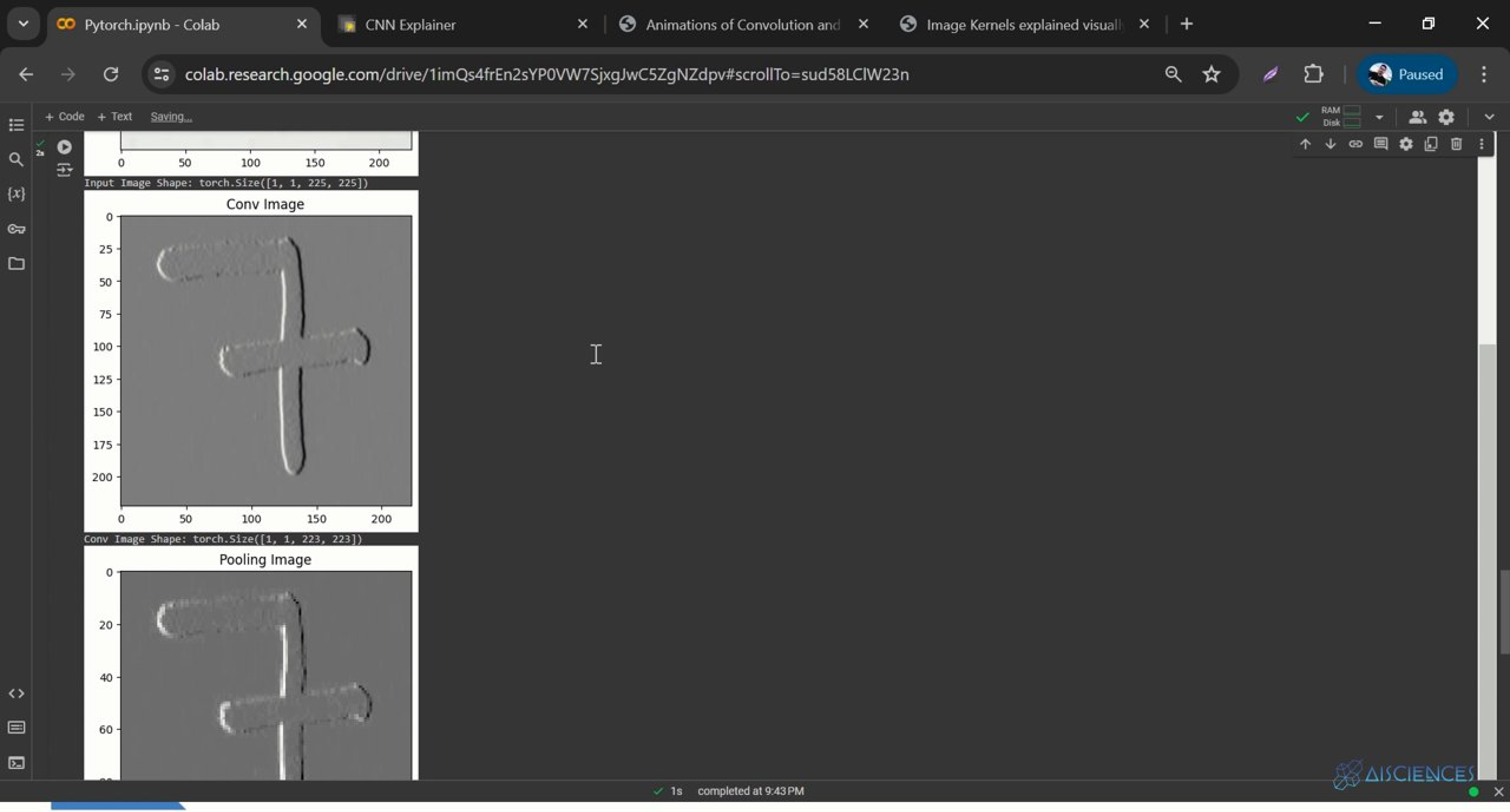
```python
    return image

# Path to the image
image_path = r"/content/drive/MyDrive/Dataset/images.jpeg"

# Load the image
input_image = load_image(image_path)
print("Input Image Shape:", input_image.shape)

# Define a convolutional layer
conv_layer = nn.Conv2d(in_channels=1, out_channels=1, kernel_size=3, stride=1, padding=0)

# Set a specific filter for the convolutional layer
conv_layer.weight = nn.Parameter(torch.tensor([[[[1, 0, -1],
                                                 [1, 0, -1],
                                                 [1, 0, -1]]]], dtype=torch.float32))

# Perform convolution
conv_output = conv_layer(input_image)
max_pooling_layer = nn.MaxPool2d(kernel_size=2, stride=2)
max_pooling_img = max_pooling_layer(conv_output)

# Display the image
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)

# Display the image
```



```
plt.imshow(input_image[0, 0].numpy(), cmap='gray')
plt.title("Input Image")
plt.show()
print("Input Image Shape:", input_image.shape)

# Display the image
plt.imshow(conv_output[0, 0].detach().numpy(), cmap='gray')
plt.title("Conv Image")
plt.show()
print("Conv Image Shape:", conv_output.shape)

# Display the image
plt.imshow(max_pooling_img[0, 0].detach().numpy(), cmap='gray')
plt.title("Pooling Image")
plt.show()
print("Pooling Image Shape:", max_pooling_img.shape)
```

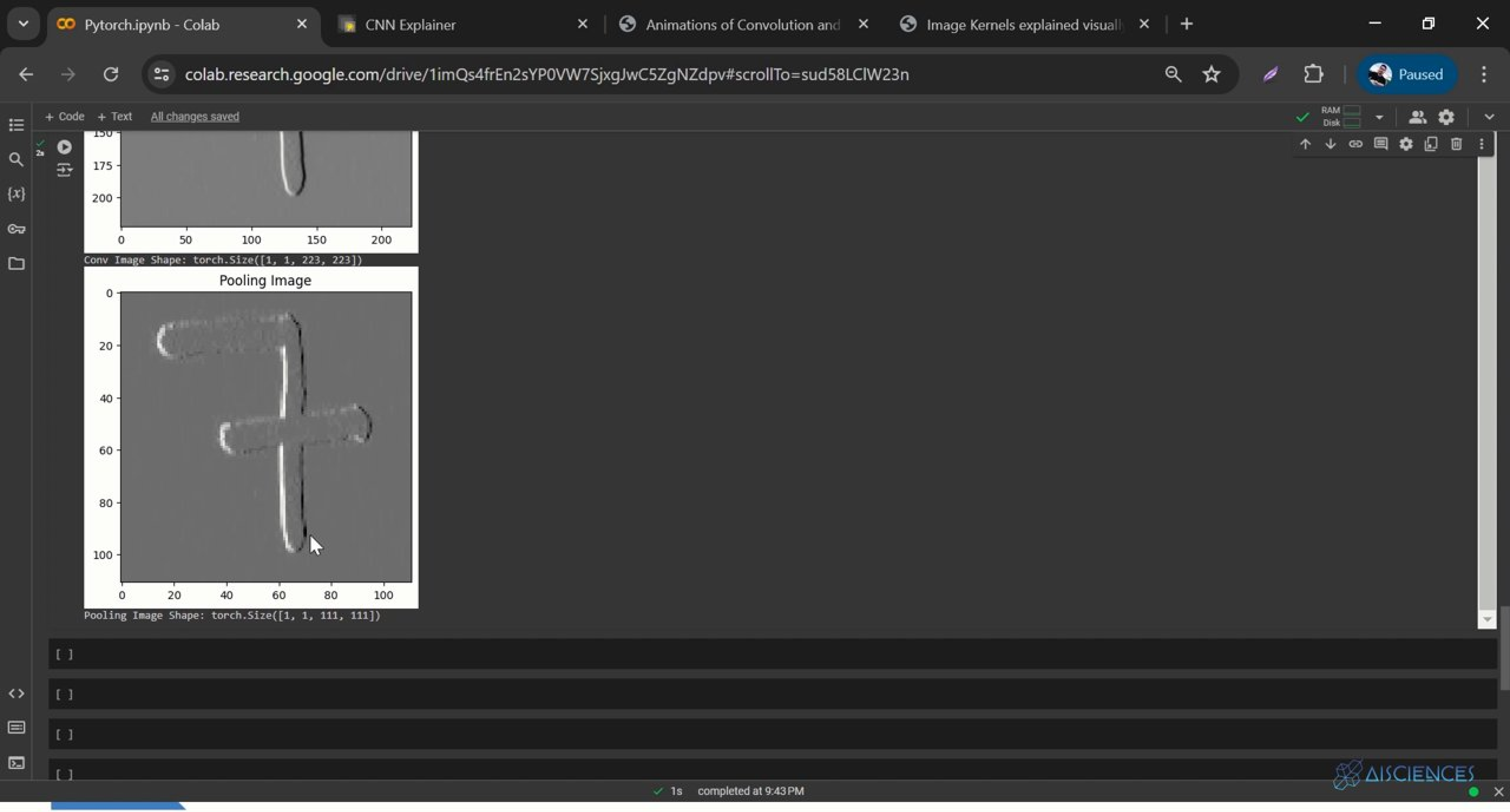
```
Input Image Shape: torch.Size([1, 1, 225, 225])
```

```
Input Image Shape: torch.Size([1, 1, 225, 225])
```



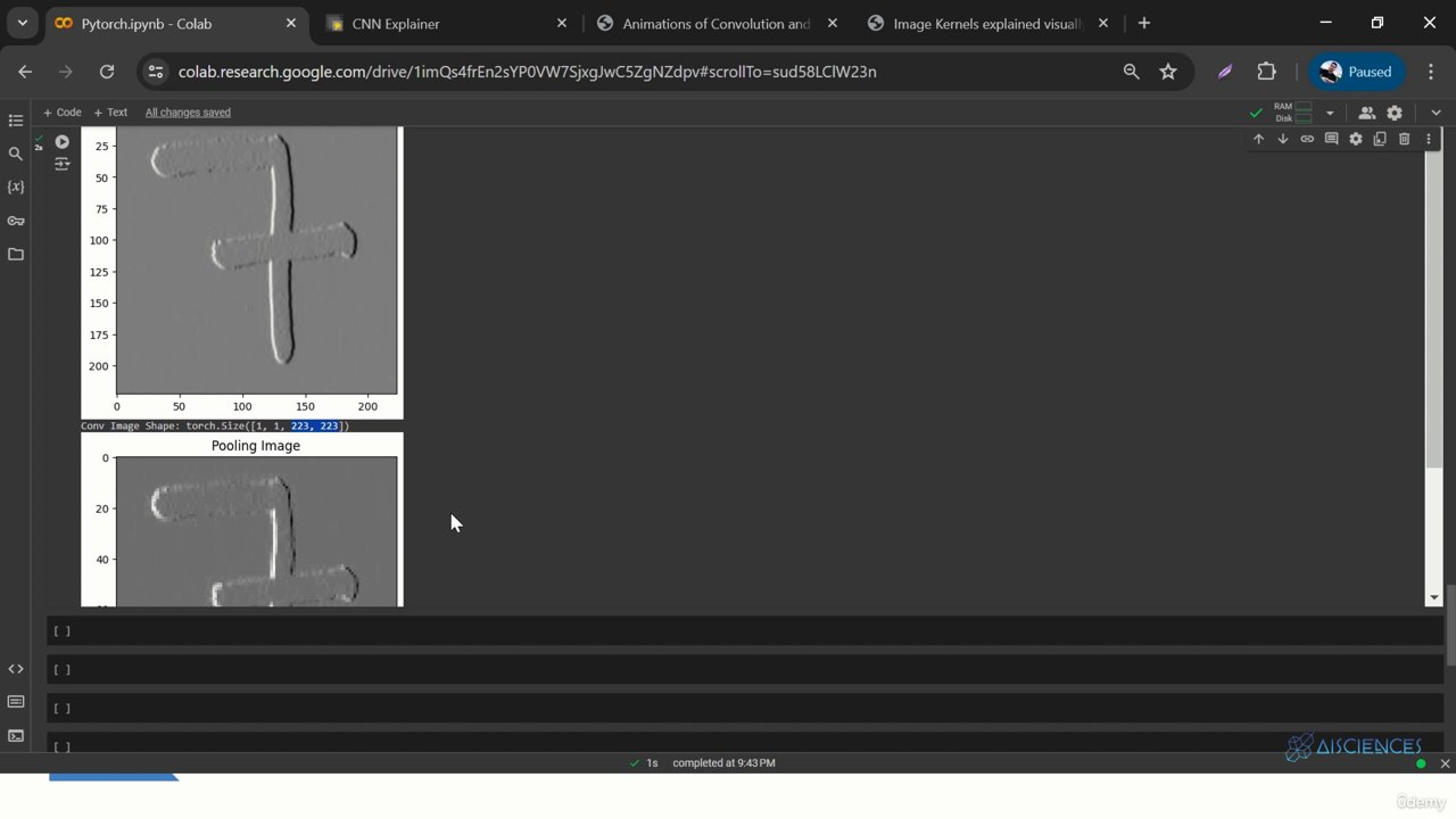
```
Input Image Shape: torch.Size([1, 1, 225, 225])
```

```
Conv Image Shape: torch.Size([1, 1, 223, 223])
```

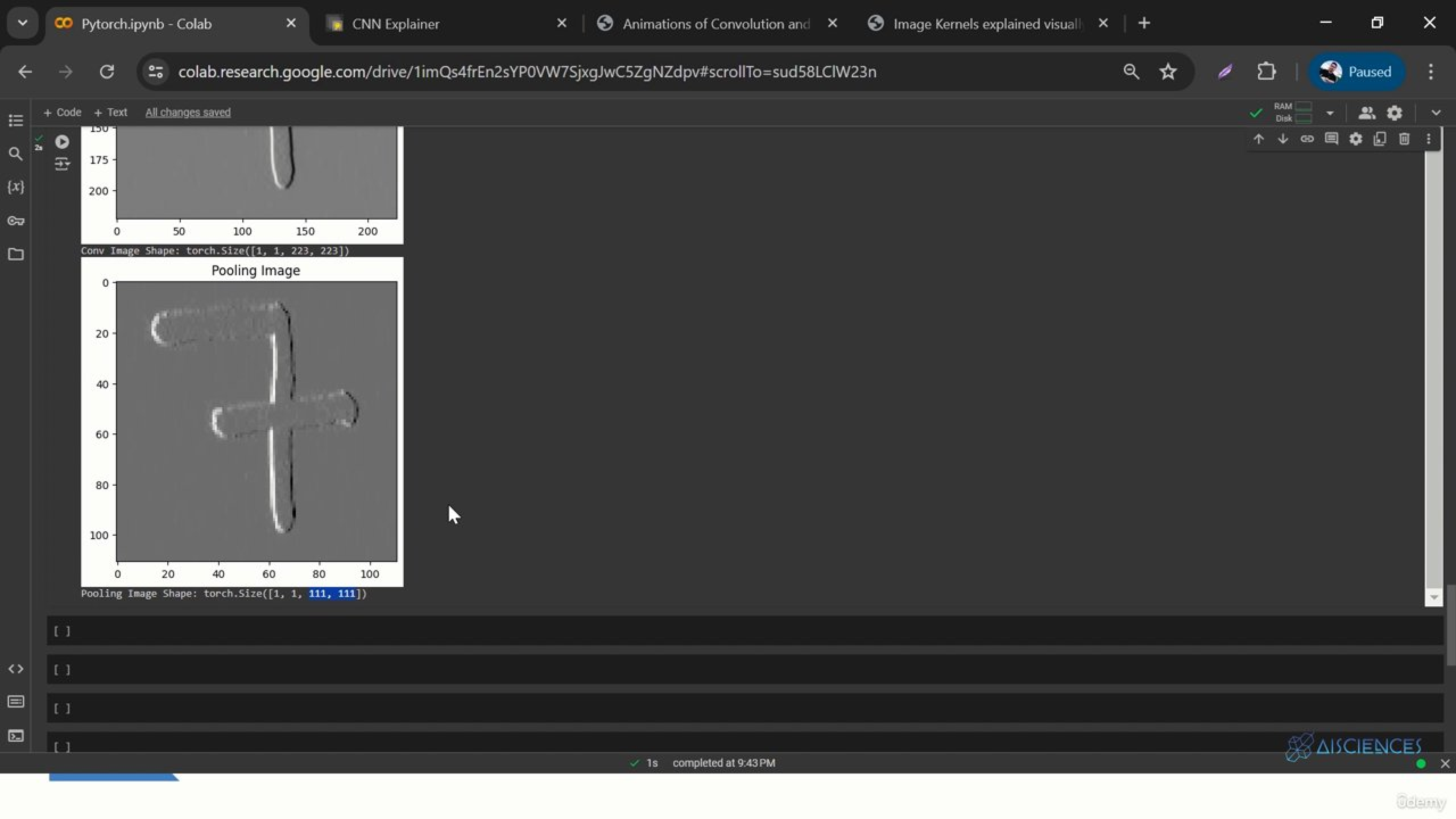


```
Conv Image Shape: torch.Size([1, 1, 223, 223])
```

```
Pooling Image Shape: torch.Size([1, 1, 111, 111])
```



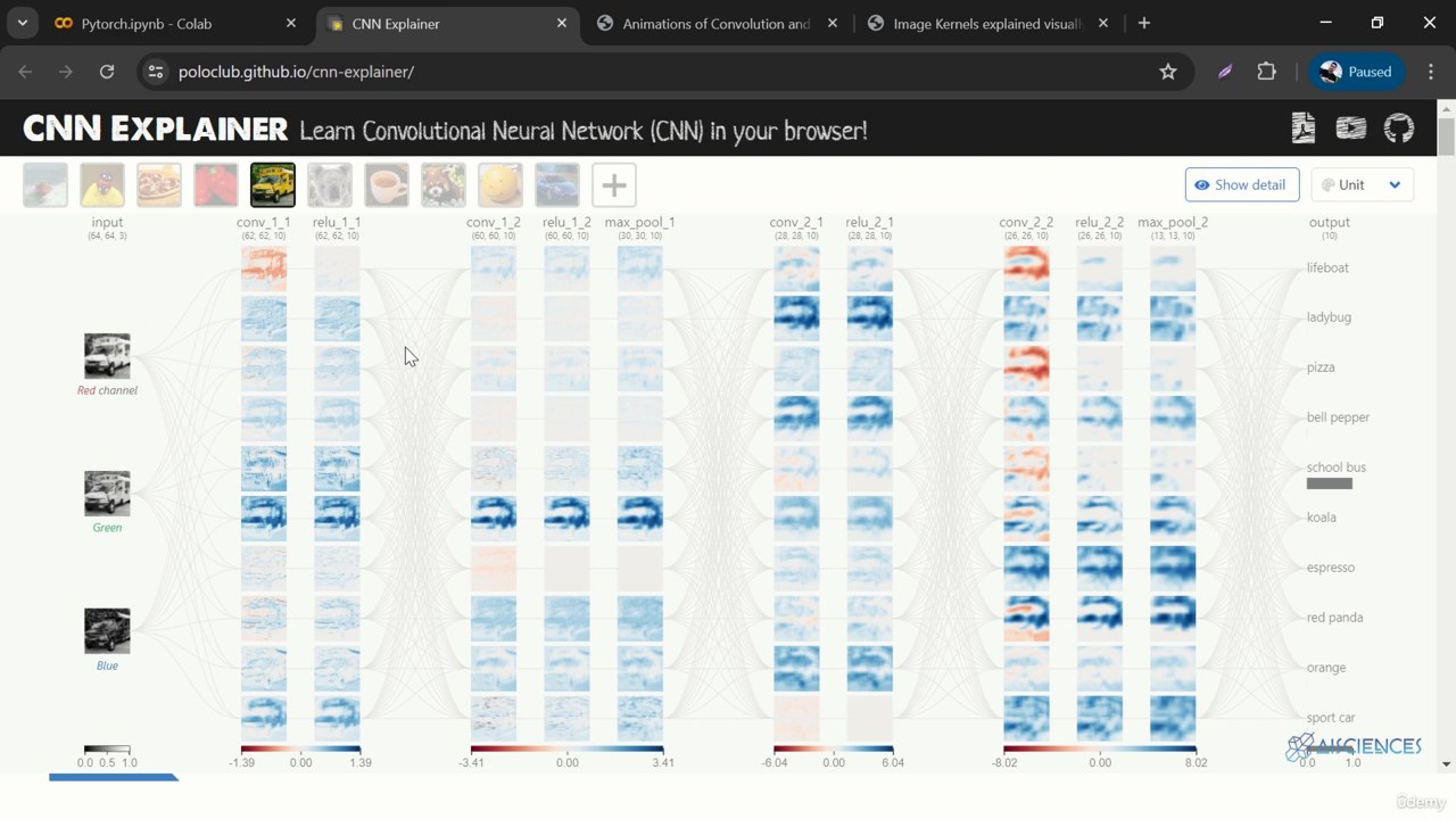
```
Conv Image Shape: torch.Size([1, 1, 223, 223])
```


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
colab.research.google.com/drive/1imQs4frEn2sYP0VW7SjxgJwC5ZgNZdpv#scrollTo=sud58LClW23n
Paused
+ Code
+ Text
All changes saved
RAM
Disk
150
175
200
0
50
100
150
200
Pooling Image
0
20
40
60
80
100
0
20
40
60
80
100
[ ]
[ ]
[ ]
[ ]

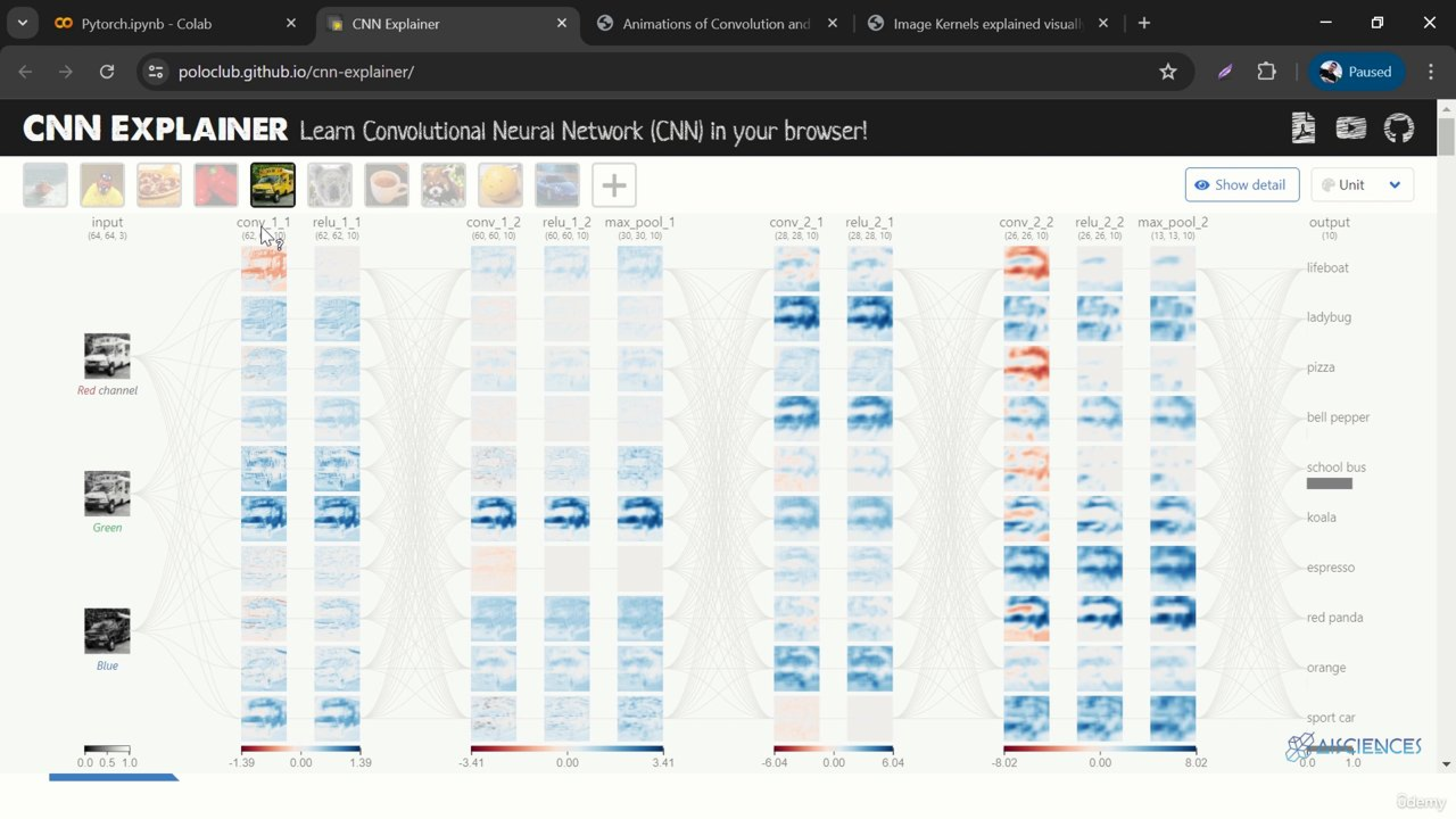
```
Conv Image Shape: torch.Size([1, 1, 223, 223])
```

```
Pooling Image Shape: torch.Size([1, 1, 111, 111])
```

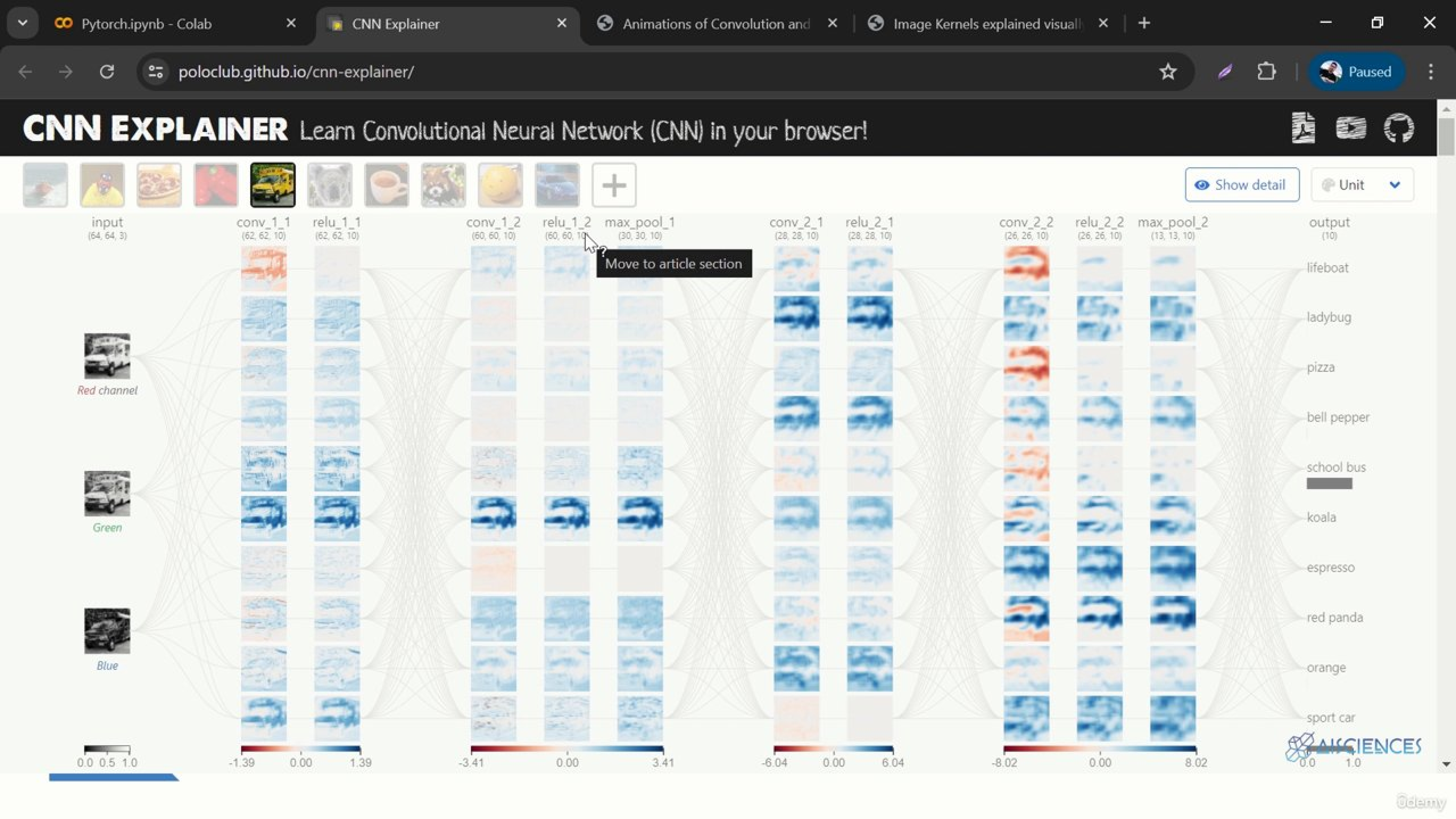
✓ 1s completed at 9:43 PM



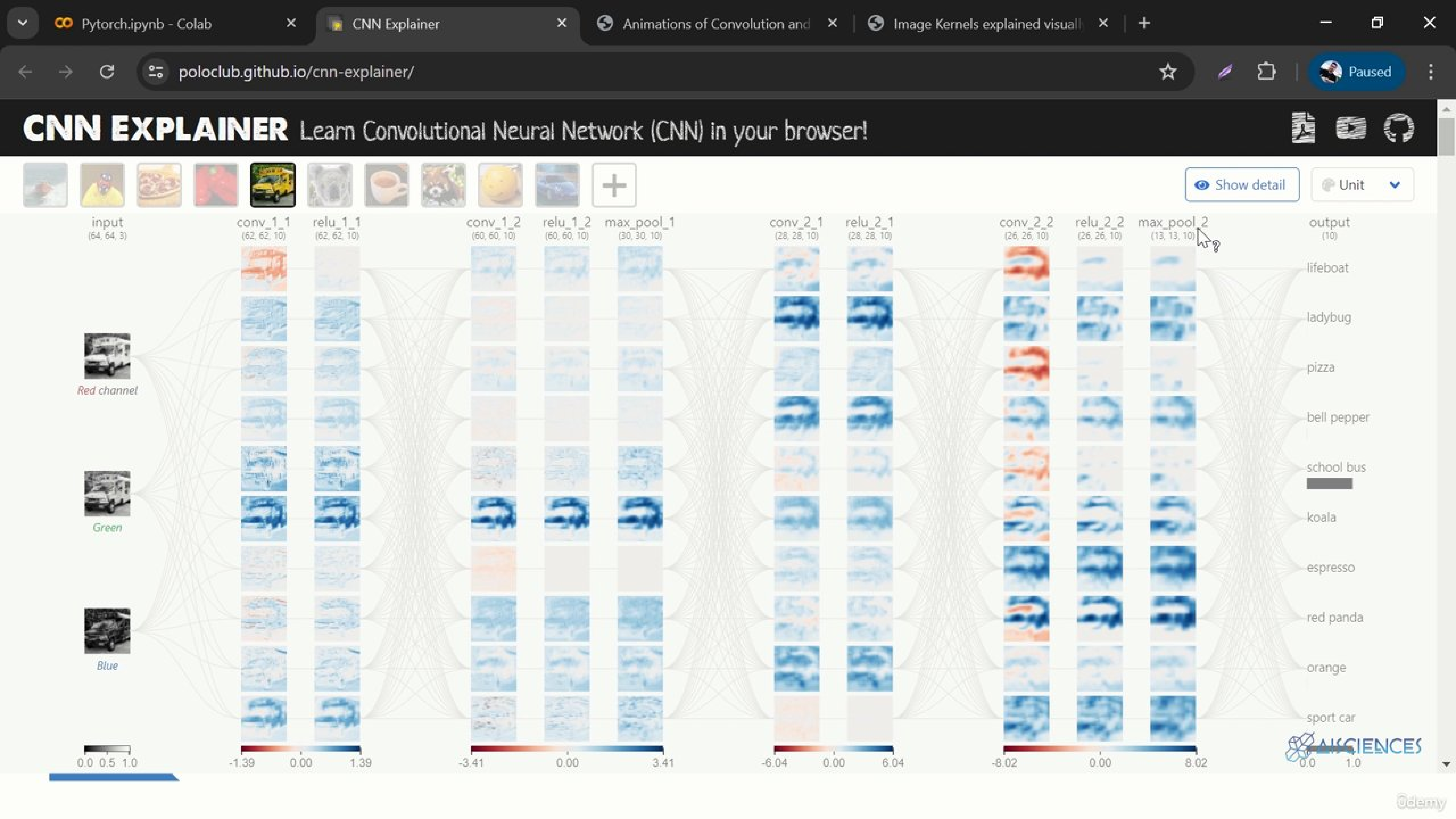
Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
poloclub.github.io/cnn-explainer/
Paused
CNN EXPLAINER Learn Convolutional Neural Network (CNN) in your browser!
Show detail
Unit
input (64, 64, 3)
conv_1_1 (62, 62, 10)
relu_1_1 (62, 62, 10)
conv_1_2 (60, 60, 10)
relu_1_2 (60, 60, 10)
max_pool_1 (30, 30, 10)
conv_2_1 (28, 28, 10)
relu_2_1 (28, 28, 10)
conv_2_2 (26, 26, 10)
relu_2_2 (26, 26, 10)
max_pool_2 (13, 13, 10)
output (10)
Red channel
Green
Blue
lifeboat
ladybug
pizza
bell pepper
school bus
koala
espresso
red panda
orange
sport car
0.0 0.5 1.0
-1.39 0.00 1.39
-3.41 0.00 3.41
-6.04 0.00 6.04
-8.02 0.00 8.02
0.0 1.0
AISCIENCES
Udemy

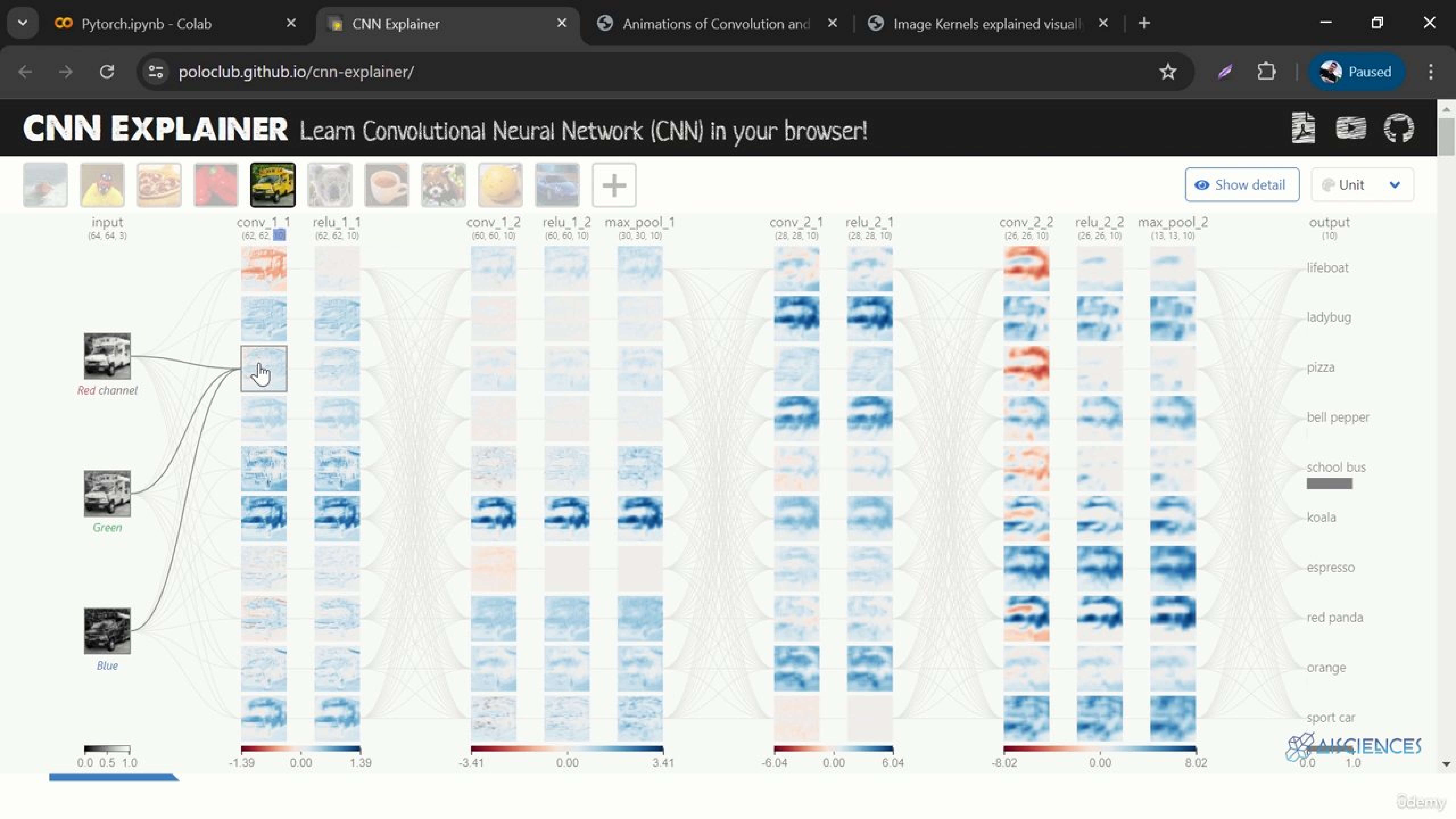


CNN EXPLAINER Learn Convolutional Neural Network (CNN) in your browser!

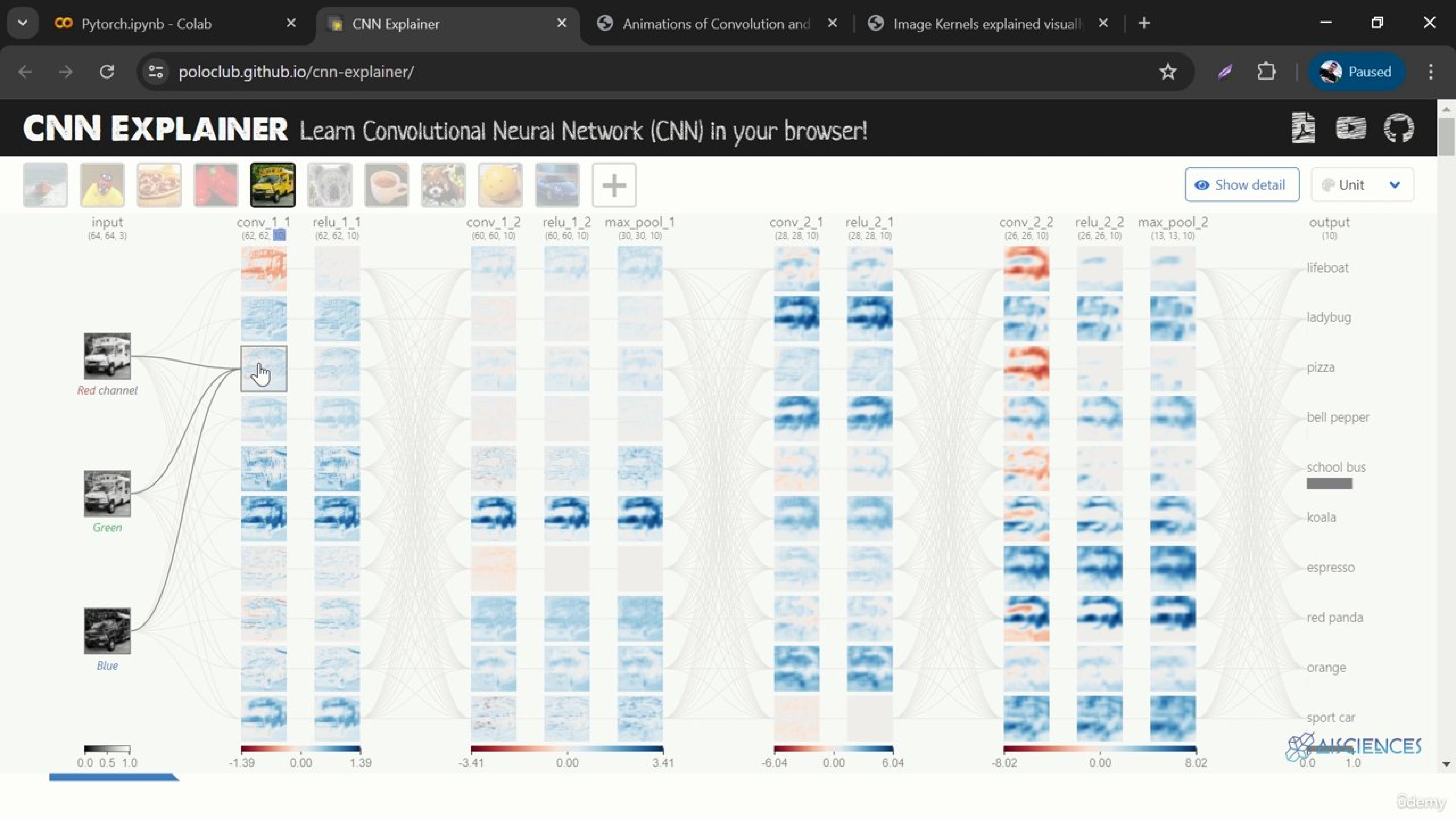


CNN EXPLAINER Learn Convolutional Neural Network (CNN) in your browser!

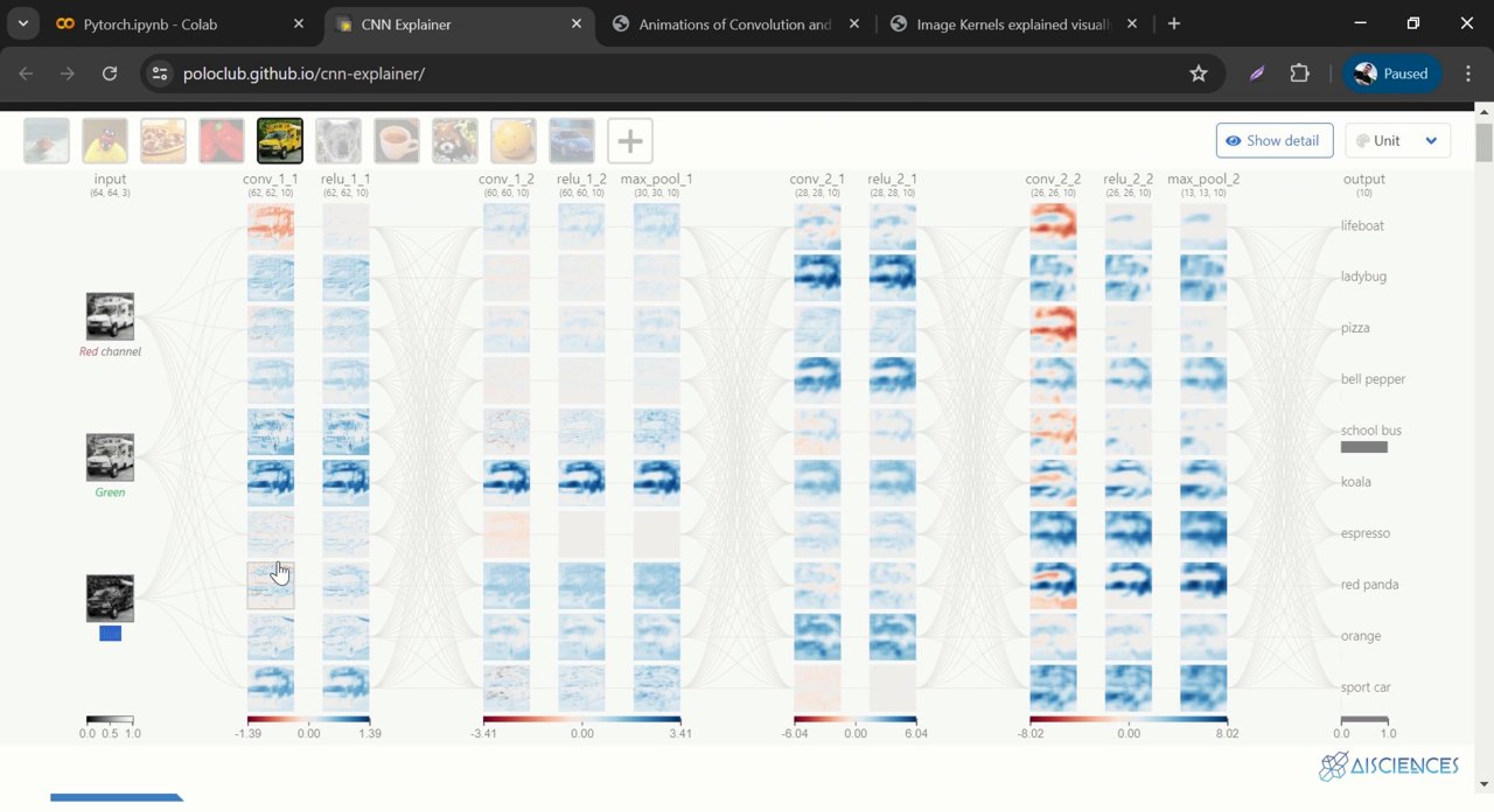

Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
poloclub.github.io/cnn-explainer/
Paused
CNN EXPLAINER Learn Convolutional Neural Network (CNN) in your browser!
Show detail
Unit
input (64, 64, 3)
conv_1_1 (62, 62, 10)
relu_1_1 (62, 62, 10)
conv_1_2 (60, 60, 10)
relu_1_2 (60, 60, 10)
max_pool_1 (30, 30, 10)
conv_2_1 (28, 28, 10)
relu_2_1 (28, 28, 10)
conv_2_2 (26, 26, 10)
relu_2_2 (26, 26, 10)
max_pool_2 (13, 13, 10)
output (10)
Red channel
Green
Blue
lifeboat
ladybug
pizza
bell pepper
school bus
koala
espresso
red panda
orange
sport car
0.0 0.5 1.0
-1.39 0.00 1.39
-3.41 0.00 3.41
-6.04 0.00 6.04
-8.02 0.00 8.02
0.0 1.0
AISCIENCES
Udemy

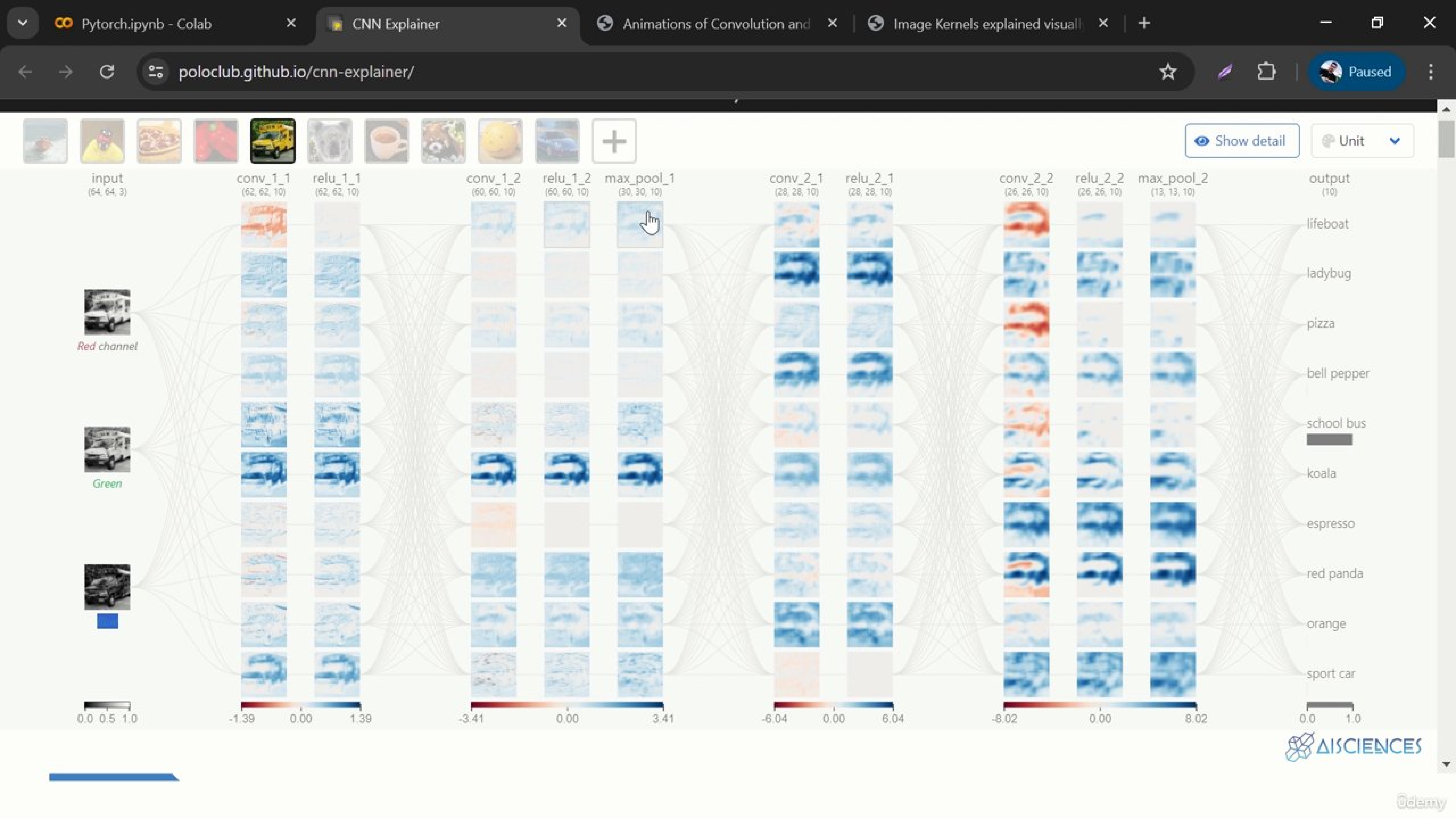


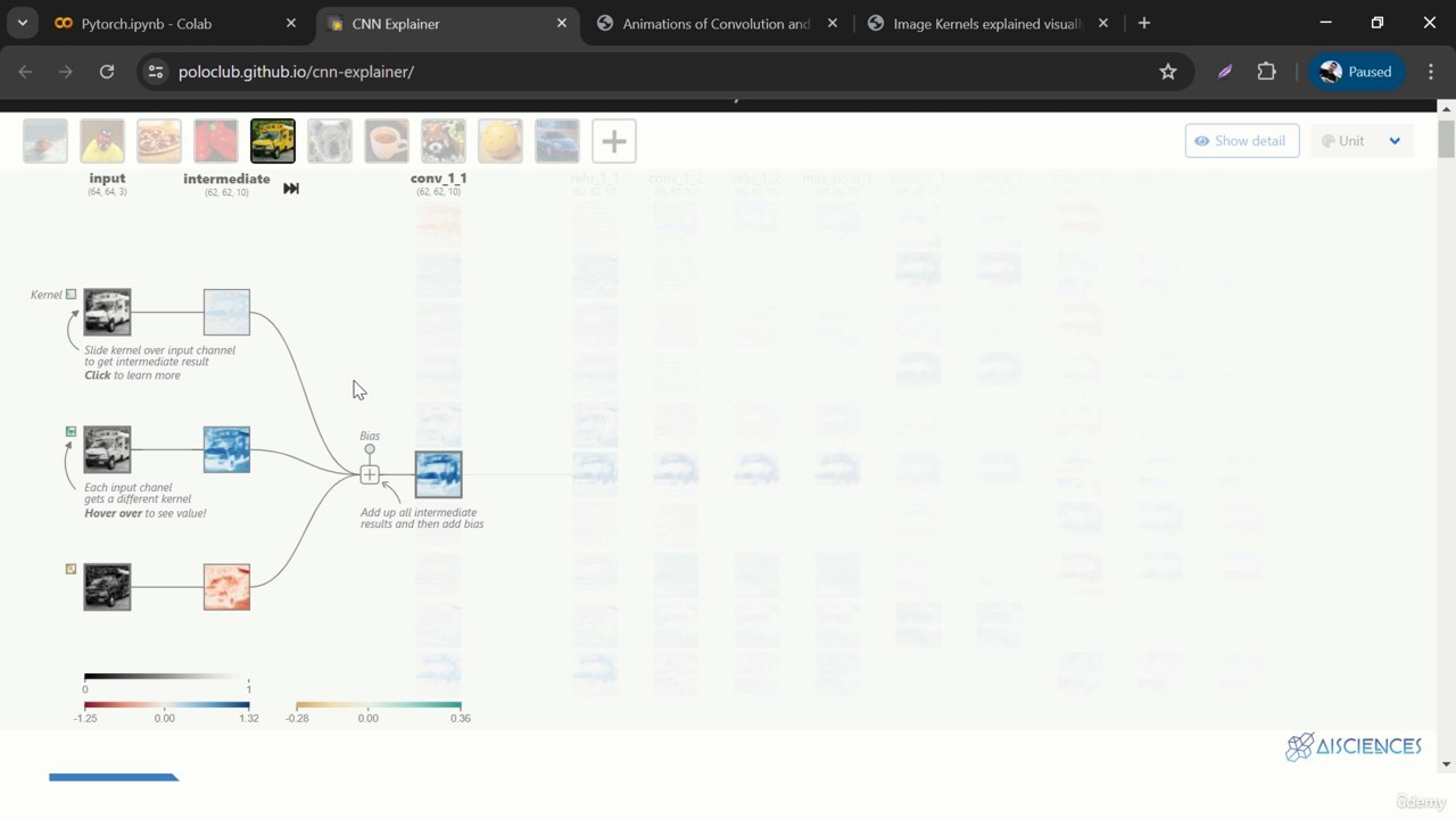
Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
poloclub.github.io/cnn-explainer/
Paused
CNN EXPLAINER Learn Convolutional Neural Network (CNN) in your browser!
Show detail
Unit
input
(64, 64, 3)
conv_1_1
(62, 62, 10)
relu_1_1
(62, 62, 10)
conv_1_2
(60, 60, 10)
relu_1_2
(60, 60, 10)
max_pool_1
(30, 30, 10)
conv_2_1
(28, 28, 10)
relu_2_1
(28, 28, 10)
conv_2_2
(26, 26, 10)
relu_2_2
(26, 26, 10)
max_pool_2
(13, 13, 10)
output
(10)
Red channel
Green
Blue
lifeboat
ladybug
pizza
bell pepper
school bus
koala
espresso
red panda
orange
sport car
0.0 0.5 1.0
-1.39 0.00 1.39
-3.41 0.00 3.41
-6.04 0.00 6.04
-8.02 0.00 8.02
0.0 1.0
AISCIENCES
Udemy



Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
poloclub.github.io/cnn-explainer/
Paused
CNN EXPLAINER Learn Convolutional Neural Network (CNN) in your browser!
Show detail
Unit
input
(64, 64, 3)
conv_1_1
(62, 62, 10)
relu_1_1
(62, 62, 10)
conv_1_2
(60, 60, 10)
relu_1_2
(60, 60, 10)
max_pool_1
(30, 30, 10)
conv_2_1
(28, 28, 10)
relu_2_1
(28, 28, 10)
conv_2_2
(26, 26, 10)
relu_2_2
(26, 26, 10)
max_pool_2
(13, 13, 10)
output
(10)
Red channel
Green
Blue
lifeboat
ladybug
pizza
bell pepper
school bus
koala
espresso
red panda
orange
sport car
0.0 0.5 1.0
-1.39 0.00 1.39
-3.41 0.00 3.41
-6.04 0.00 6.04
-8.02 0.00 8.02
0.0 1.0
AISCIENCES
Udemy


Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
poloclub.github.io/cnn-explainer/
Paused
Show detail
Unit
input
(64, 64, 3)
conv_1_1
(62, 62, 10)
relu_1_1
(62, 62, 10)
conv_1_2
(60, 60, 10)
relu_1_2
(60, 60, 10)
max_pool_1
(30, 30, 10)
conv_2_1
(28, 28, 10)
relu_2_1
(28, 28, 10)
conv_2_2
(26, 26, 10)
relu_2_2
(26, 26, 10)
max_pool_2
(13, 13, 10)
output
(10)
Red channel
Green
lifeboat
ladybug
pizza
bell pepper
school bus
koala
espresso
red panda
orange
sport car
0.0 0.5 1.0
-1.39 0.00 1.39
-3.41 0.00 3.41
-6.04 0.00 6.04
-8.02 0.00 8.02
0.0 1.0
AISCIENCES

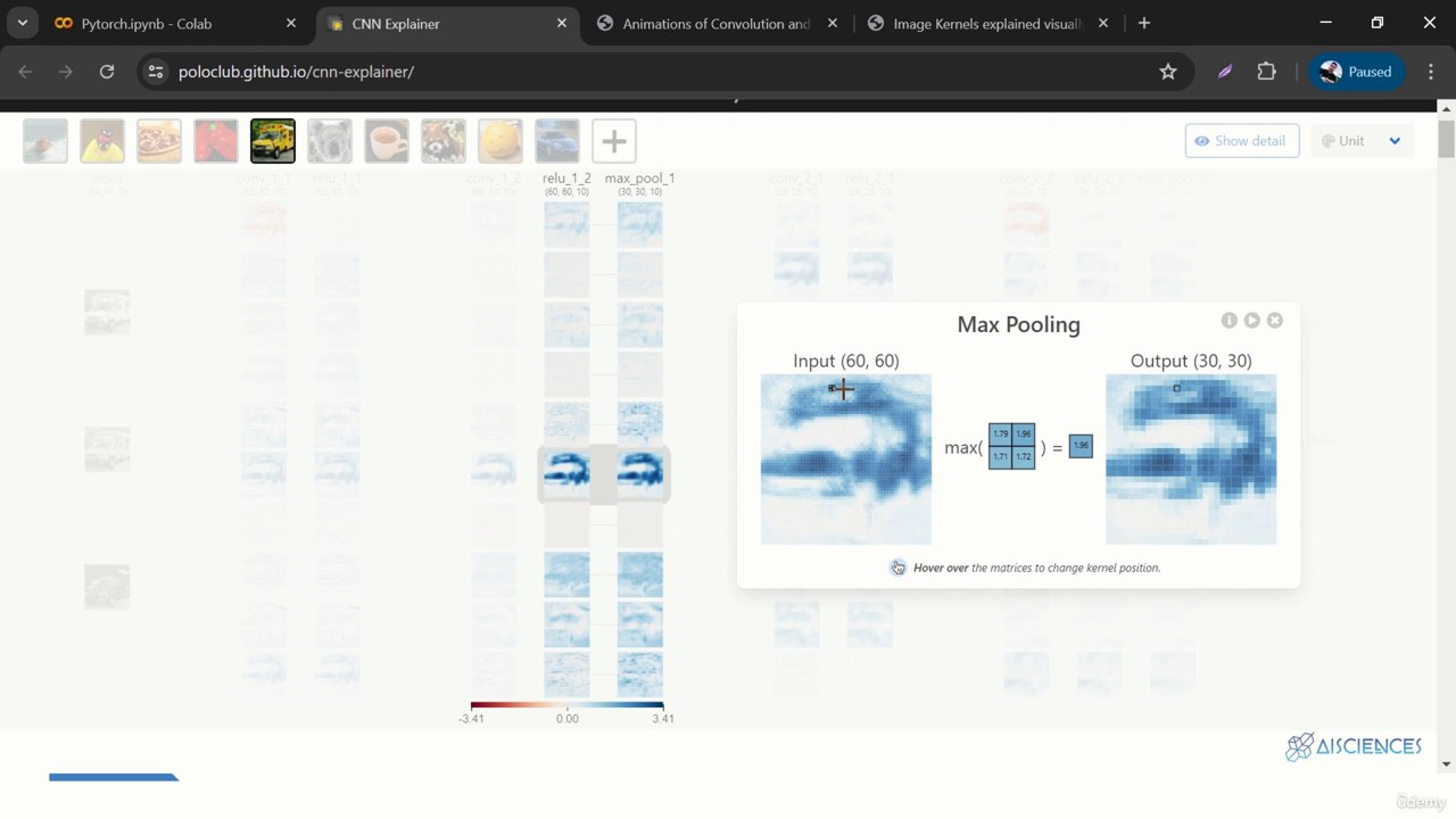

Pytorch.ipynb - Colab
CNN Explainer
Animations of Convolution and
Image Kernels explained visuall
poloclub.github.io/cnn-explainer/
Paused
Show detail
Unit
input
(64, 64, 3)
conv_1_1
(62, 62, 10)
relu_1_1
(62, 62, 10)
conv_1_2
(60, 60, 10)
relu_1_2
(60, 60, 10)
max_pool_1
(30, 30, 10)
conv_2_1
(28, 28, 10)
relu_2_1
(28, 28, 10)
conv_2_2
(26, 26, 10)
relu_2_2
(26, 26, 10)
max_pool_2
(13, 13, 10)
output
(10)
Red channel
Green
lifeboat
ladybug
pizza
bell pepper
school bus
koala
espresso
red panda
orange
sport car
0.0 0.5 1.0
-1.39 0.00 1.39
-3.41 0.00 3.41
-6.04 0.00 6.04
-8.02 0.00 8.02
0.0 1.0
AISCIENCES
Udemy



Slide kernel over input channel to get intermediate result
Click to learn more

Each input chanel gets a different kernel
Hover over to see value!

Add up all intermediate results and then add bias



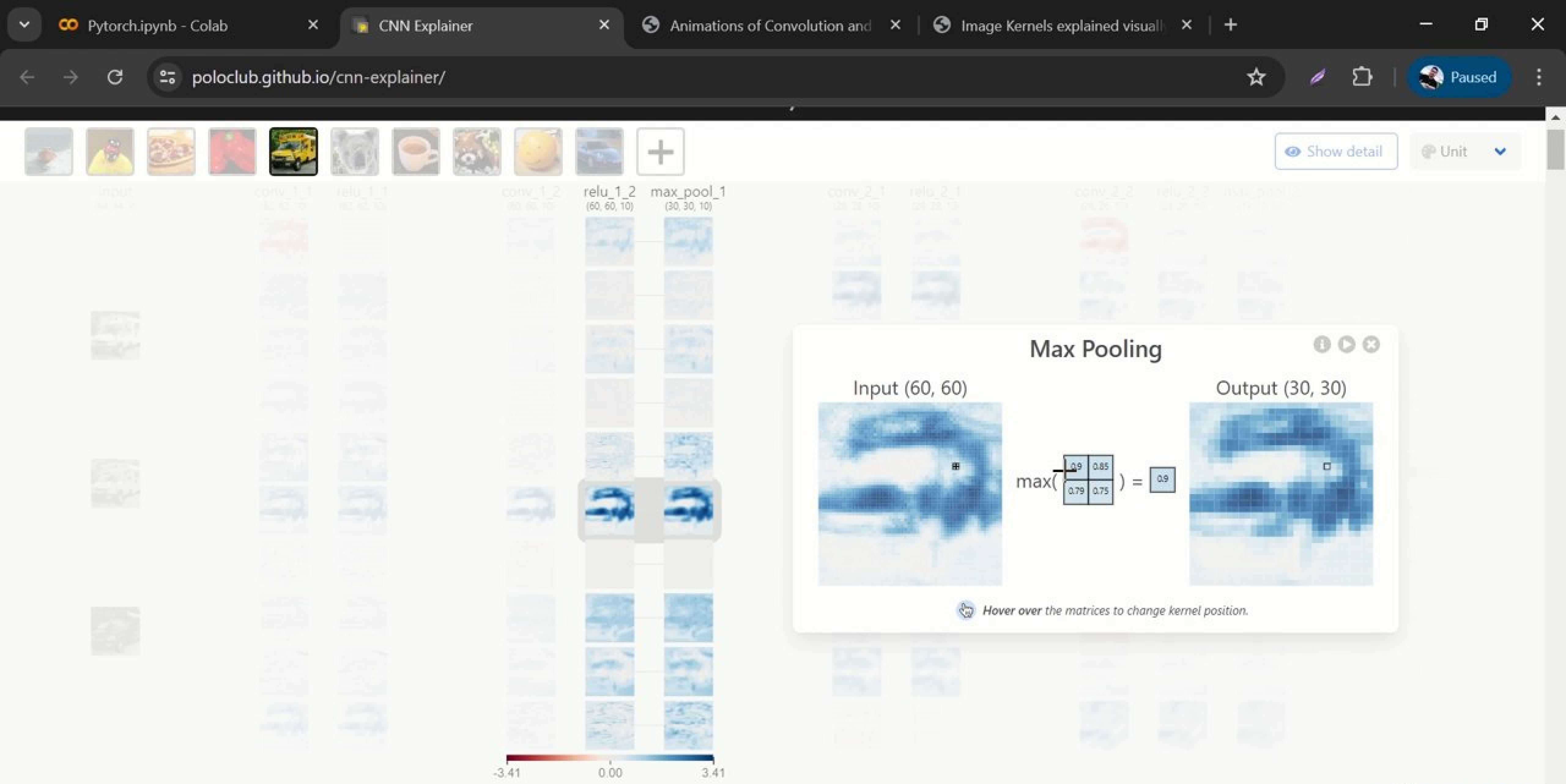
Max Pooling

Input (60, 60)

Output (30, 30)

$$\max\left(\begin{array}{|c|c|}\hline 1.79 & 1.96 \\ \hline 1.71 & 1.72 \\ \hline\end{array}\right) = \boxed{1.96}$$

Hover over *the matrices to change kernel position.*



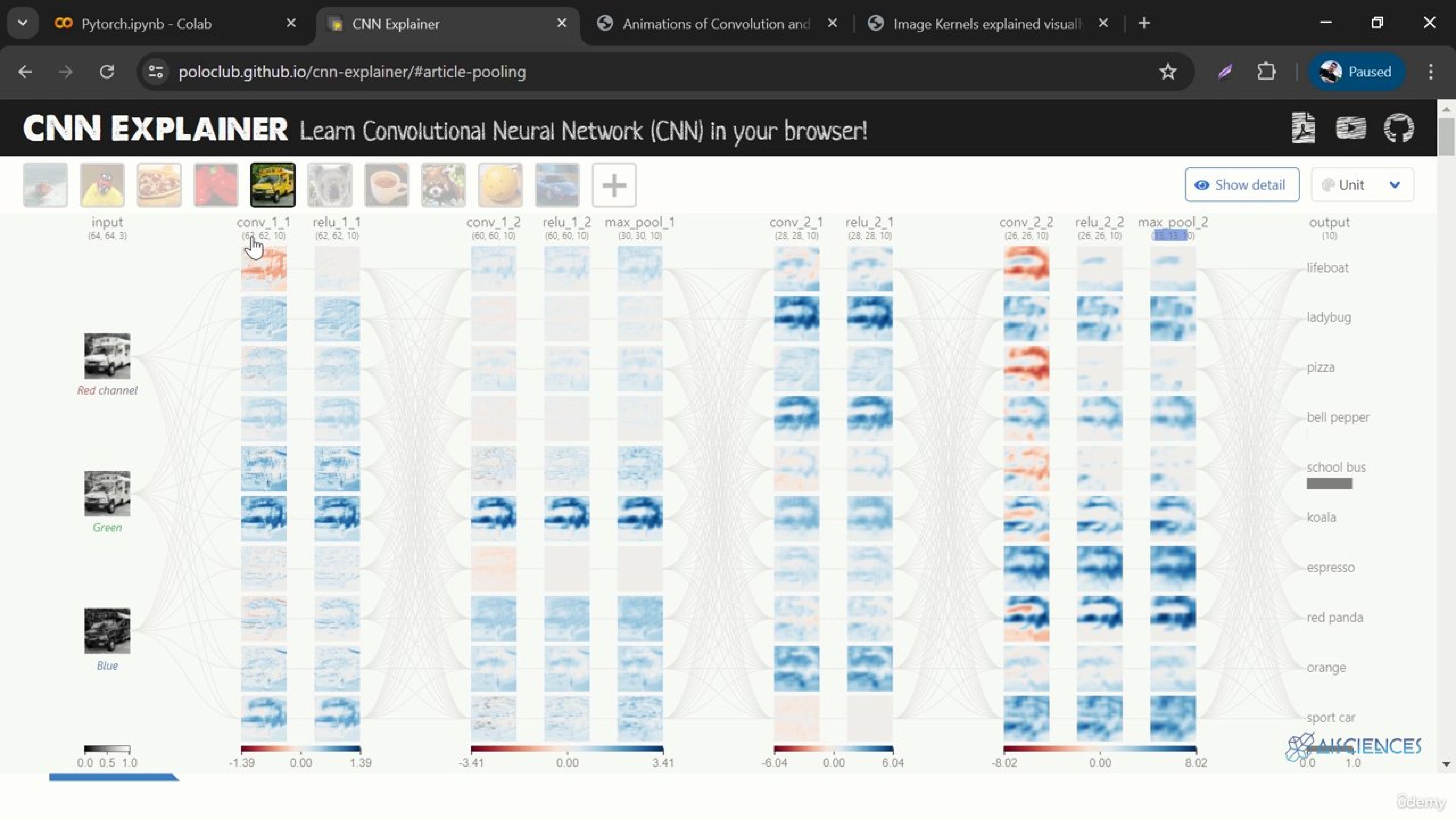
Max Pooling

Input (60, 60)

Output (30, 30)

$$\max\left(\begin{bmatrix} 0.9 & 0.85 \\ 0.79 & 0.75 \end{bmatrix}\right) = \boxed{0.9}$$

Hover over *the matrices to change kernel position.*



poloclub.github.io/cnn-explainer/#article-pooling

CNN EXPLAINER Learn Convolutional Neural Network (CNN) in your browser!

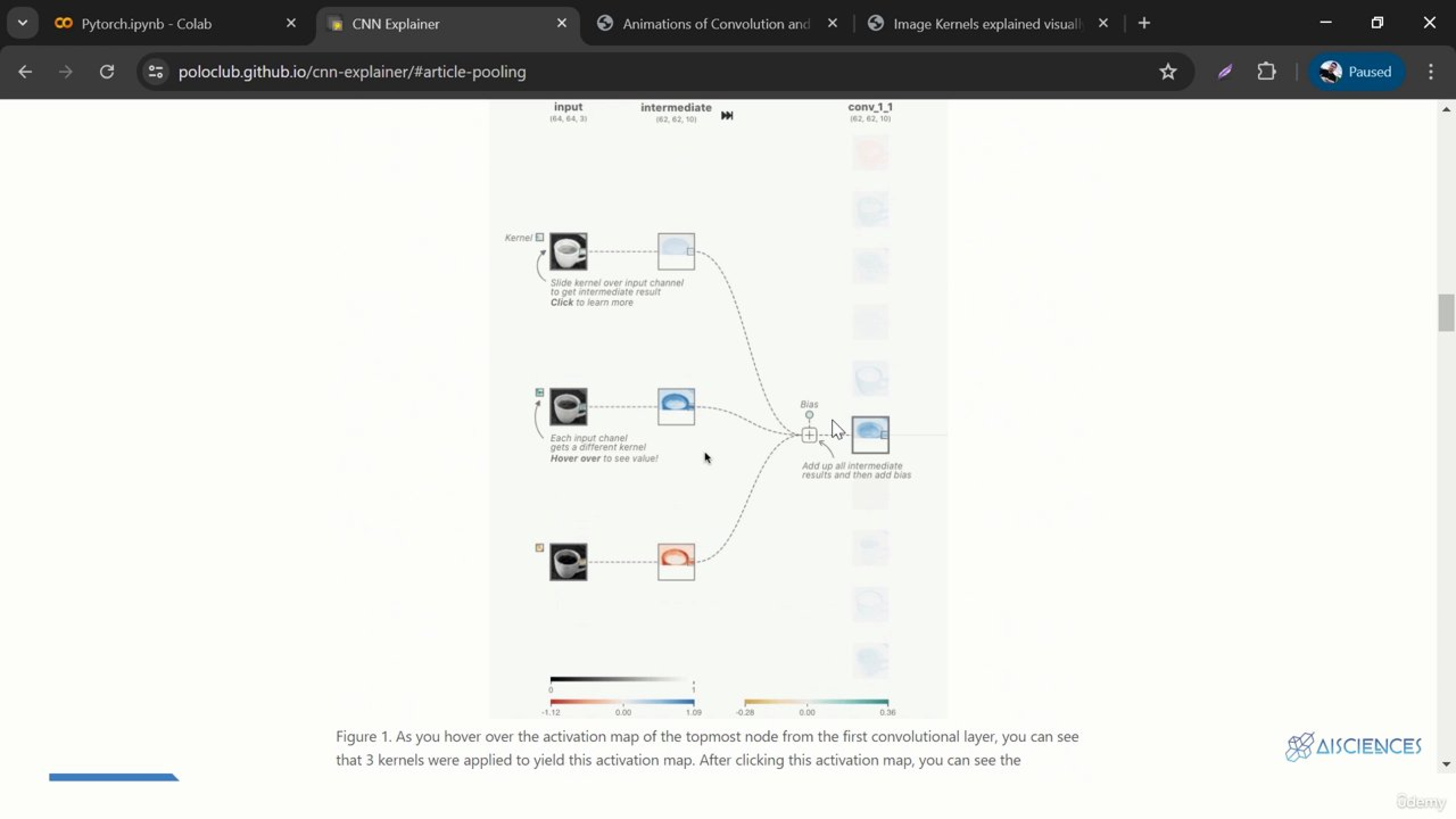


Figure 1. As you hover over the activation map of the topmost node from the first convolutional layer, you can see that 3 kernels were applied to yield this activation map. After clicking this activation map, you can see the

MAX POOLING

Input

$$\begin{bmatrix} 6 & 3 & 2 \\ 5 & 2 & 1 \\ 4 & 1 & 0 \end{bmatrix}$$

Pooling matrix
2x2

Output

$$\begin{bmatrix} 6 & 2 \\ 4 & 0 \end{bmatrix}$$

CONVOLUTION

Input

$$\begin{bmatrix} 1 & 2 & 0 & 1 & 3 \\ 4 & 1 & 0 & 2 & 1 \\ 2 & 3 & 1 & 0 & 1 \\ 1 & 2 & 0 & 1 & 3 \\ 3 & 1 & 2 & 3 & 0 \end{bmatrix}$$

Kernal

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Output

$$\begin{bmatrix} 6 & 3 & 2 \\ 5 & 2 & 1 \\ 4 & 1 & 0 \end{bmatrix}$$

Output(1,2) = (2×1)+(0×0)+(1×−1)+(1×1)+(0×0)+(2×−1)+(3×1)+(1×0)+(0×−1)

=2+0−1+1+0−2+3+0−0

=3