МИНОБРНАУКИРОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА— Российский технологический университет»
РТУМИРЭА

Институт кибербезопасности и цифровых технологий направление 10.04.01

«Информационная безопасность»

Кафедра КБ-4«Интеллектуальные системы информационной безопасности»

Лабораторная работа №3

По дисциплине
«Анализ защищенности систем искусственного интеллекта»

Выполнил:

Суслов Антон Константинович

Группа: ББМО-02-22

Москва 2023

Установка нужных инструментов и импорт библиотек

```
!pip install tf-keras-vis
```

```
Collecting tf-keras-vis
  Downloading tf_keras_vis-0.8.6-py3-none-any.whl (52 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 52.1/52.1 kB 1.2 MB/s eta 0:
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-pa
Collecting deprecated (from tf-keras-vis)
  Downloading Deprecated-1.2.14-py2.py3-none-any.whl (9.6 kB)
Requirement already satisfied: imageio in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist
Requirement already satisfied: wrapt<2,>=1.10 in /usr/local/lib/python3.10
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-pac
Installing collected packages: deprecated, tf-keras-vis
Successfully installed deprecated-1.2.14 tf-keras-vis-0.8.6
```

```
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import cm
import tensorflow as tf
from tf_keras_vis.gradcam import Gradcam
from tf_keras_vis.saliency import Saliency
from tf_keras_vis.utils.scores import CategoricalScore
from tf_keras_vis.utils.model_modifiers import ReplaceToLinear
from tf_keras_vis.gradcam_plus_plus import GradcamPlusPlus
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.applications.vgg16 import preprocess_input
from tensorflow.keras.applications.vgg16 import VGG16 as Model
```

4 разных изображения

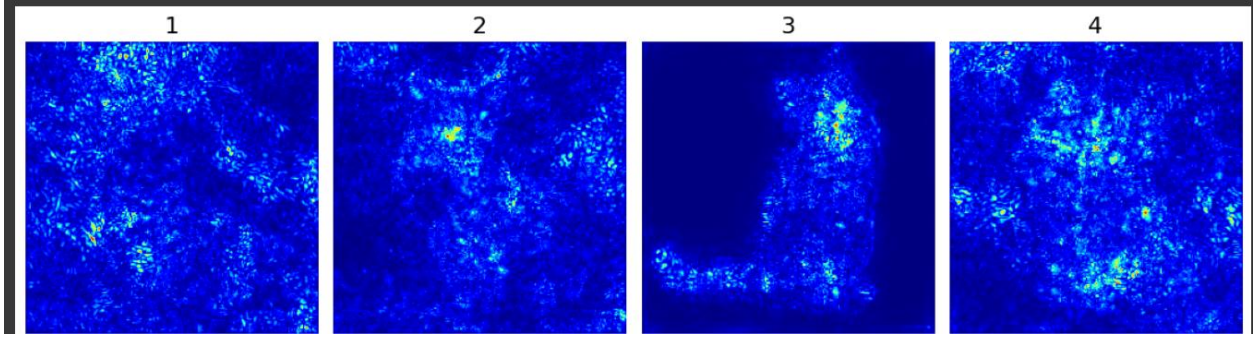
```
imgPathList = ['1.jpg', '2.jpg', '3.jpg', '4.jpg']

imgArr = np.asarray([np.array(load_img(imgPath, target_size=(224, 224))) for imgPath in imgPathList])

X = preprocess_input(imgArr)

f, ax = plt.subplots(nrows=1, ncols=4, figsize=(12, 4))
for i, title in enumerate(imgTitleList):
    ax[i].set_title(title, fontsize=16)
    ax[i].imshow(imgArr[i])
    ax[i].axis('off')
plt.tight_layout()
plt.show()
```

Устанавливаем классы изображений

```
replace2linear = ReplaceToLinear()
def model_modifier_function(cloned_model):
    cloned_model.layers[-1].activation = tf.keras.activations.linear

score = CategoricalScore([41, 42, 62, 63])
def score_function(output):
    return (output[0][41], output[1][42], output[2][62], output[3][63])
```

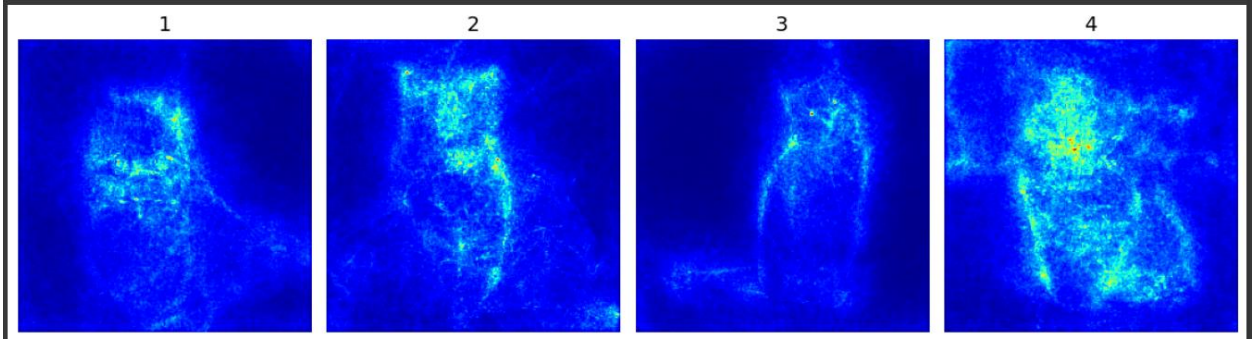
Отображение карты значимости

```
saliency = Saliency(model,model_modifier=replace2linear,clone=True)
mapList = saliency(score, X)

f, ax = plt.subplots(nrows=1, ncols=4, figsize=(12, 4))
for i, title in enumerate(imgTitleList):
    ax[i].set_title(title, fontsize=16)
    ax[i].imshow(mapList[i], cmap='jet')
    ax[i].axis('off')

plt.tight_layout()
plt.show()
```



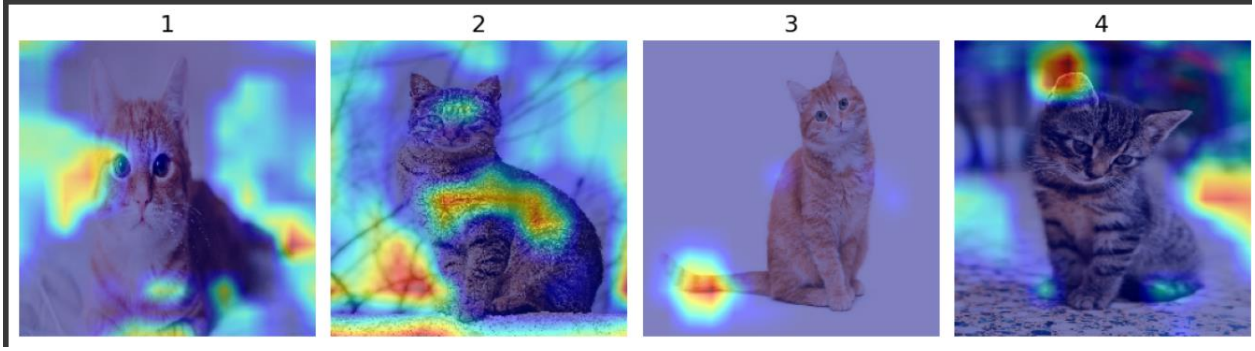
Удаление шума

```
mapList = saliency(score,X,smooth_samples=20,smooth_noise=0.20)
f, ax = plt.subplots(nrows=1, ncols=4, figsize=(12, 4))

for i, title in enumerate(imgTitleList):
    ax[i].set_title(title, fontsize=14)
    ax[i].imshow(mapList[i], cmap='jet')
    ax[i].axis('off')

plt.tight_layout()
plt.show()
```



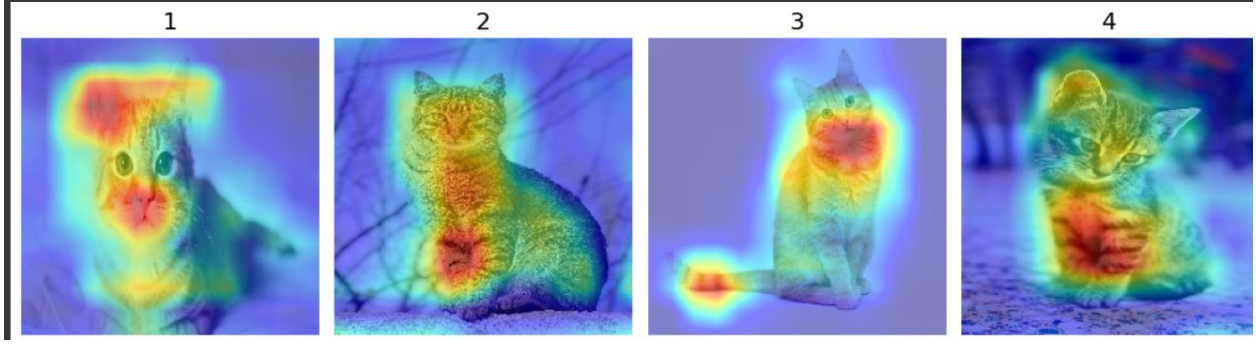
Использование GradCam

```
gradcam = Gradcam(model,model_modifier=replace2linear,clone=True)
mapList = gradcam(score,X,penultimate_layer=-1)
f, ax = plt.subplots(nrows=1, ncols=4, figsize=(12, 4))

for i, title in enumerate(imgTitleList):
    heatmap = np.uint8(cm.jet(mapList[i])[..., :4] * 255)
    ax[i].set_title(title, fontsize=16)
    ax[i].imshow(imgArr[i])
    ax[i].imshow(heatmap, cmap='jet', alpha=0.5)
    ax[i].axis('off')

plt.tight_layout()
plt.show()
```



Использование GradCam++

```
gradcam = GradcamPlusPlus(model,model_modifier=replace2linear,clone=True)
mapList = gradcam(score,X,penultimate_layer=-1)
f, ax = plt.subplots(nrows=1, ncols=4, figsize=(12, 4))

for i, title in enumerate(imgTitleList):
    heatmap = np.uint8(cm.jet(mapList[i])[..., :4] * 255)
    ax[i].set_title(title, fontsize=16)
    ax[i].imshow(imgArr[i])
    ax[i].imshow(heatmap, cmap='jet', alpha=0.5)
    ax[i].axis('off')

plt.tight_layout()
plt.show()
```



Вывод: Модификаторы None, guided, relu влияют на способ вычисления градиентов и на результаты визуализации. Применение функций активации relu в модификаторах может изменять визуализацию и помогать понять, какие части модели более активны.