



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ Информатика, искусственный интеллект и системы управления

КАФЕДРА Системы обработки информации и управления

## ОТЧЕТ

### К ДОМАШНЕМУ ЗАДАНИЮ № 3

По дисциплине «Методы поддержки принятия решений»

### МЕТОДЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ МЕТОДОВ ВЫЧИСЛИТЕЛЬНОГО ИНТЕЛЛЕКТА

Студент ИУ5-716  
(Группа)

\_\_\_\_\_  
(Подпись, дата) Д.К.Бондаренко  
(И.О.Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата) А.Н. Нардид  
(И.О.Фамилия)

Оценка \_\_\_\_\_

## Задание

Необходимо создать web-приложение для классификации изображений с использованием предобученной модели на основе датасета cifar100. При выполнении задания Вы должны использовать 3-и класса из CIFAR100.

### Исходные данные:

- 1) Класс 3: медведь;
- 2) Класс 33: лес;
- 3) Класс 63: дикобраз;
- 4) Модель: `cifar100_mobile`.

## Выполнение работы

### Часть 1. Сохранение модели в Google Colab

Были импортированы необходимые библиотеки.

```
%pip install onnx
import torch

Collecting onnx
  Downloading onnx-1.15.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (15.7 MB)
    15.7/15.7 MB 63.0 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from onnx) (1.23.5)
Requirement already satisfied: protobuf>=3.20.2 in /usr/local/lib/python3.10/dist-packages (from onnx) (3.20.3)
Installing collected packages: onnx
Successfully installed onnx-1.15.0
```

Был создан девайс.

```
[ ] device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

Была загружена модель в соответствии с вариантом.

```
[ ] model = torch.hub.load("chenyafo/pytorch-cifar-models",
    "cifar100_mobilenetv2_x0_5",
    # 'cifar100_resnet20',
    pretrained=True)

/usr/local/lib/python3.10/dist-packages/torch/hub.py:294: UserWarning: You are about to download and run co
warnings.warn(
Downloading: "https://github.com/chenyafo/pytorch-cifar-models/zipball/master" to /root/.cache/torch/hub/m
Downloading: "https://github.com/chenyafo/pytorch-cifar-models/releases/download/mobilenetv2/cifar100_mobi
100%|██████████| 3.29M/3.29M [00:00<00:00, 76.8MB/s]
```

Модель была загружена на девайс.

```
[ ] model.to(device)

MobileNetV2(
  (features): Sequential(
    (0): ConvBNActivation(
      (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNActivation(
          (0): Conv2d(16, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=16, bias=False)
          (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2d(16, 8, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (2): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (2): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNActivation(
          (0): Conv2d(8, 48, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
      )
    )
  )
)
```

Модель была экспортирована и сохранена в формате onnx.

```
[ ] x = torch.randn(1, 3, 32, 32, requires_grad=True).to(device)

onnx.export(model, # модель
            x, # входной тензор (или кортеж нескольких тензоров)
            "cifar100_CNN_RESNET20.onnx", # куда сохранить (либо путь к файлу либо fileObject)
            export_params=True, # сохраняет веса обученных параметров внутри файла модели
            opset_version=9, # версия ONNX
            do_constant_folding=True, # следует ли выполнять укорачивание констант для оптимизации
            input_names = ['input'], # имя входного слоя
            output_names = ['output'], # имя выходного слоя
            dynamic_axes={'input' : {0 : 'batch_size'}, # динамические оси, в данном случае только размер пакета
                          'output' : {0 : 'batch_size'}})
```

## Часть 2. Web-приложение классификации изображений

Был создан проект Django в IDE Pycharm. В корне проекта была создана папка “media” для последующего сохранения изображений и файлов формата ONNX. Внутри папки “media” для этого были созданы папки "images" и "models".

Файл settings.py:

```
"""
Django settings for MPPR_DZ3 project.

Generated by 'django-admin startproject' using Django 4.2.7.

For more information on this file, see
```

```

https://docs.djangoproject.com/en/4.2/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.2/ref/settings/
"""

from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-
me$=@^sh@ck%7v@7_g@bso101(=ih#3()h2w0e5a762i&aemm4'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'MPPR_DZ3_app.apps.MpprDz3AppConfig',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'MPPR_DZ3.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates']
    },
    {
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

```

```

    },
]

WSGI_APPLICATION = 'MPPR_DZ3.wsgi.application'

# Database
# https://docs.djangoproject.com/en/4.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/4.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/4.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

MEDIA_URL = '/media/'
MEDIA_ROOT =

```

```
os.path.join(os.path.dirname(os.path.dirname(os.path.abspath(__file__))),
'media')
```

Файл views.py:

```
from django.shortcuts import render
from django.core.files.storage import FileSystemStorage
import onnxruntime
import numpy as np
from PIL import Image
from io import BytesIO
import base64
from torchvision import transforms

imageClassList = {3: 'bear', 33: 'forest', 63: 'porcupine'} # Сюда указать
КЛАССЫ

def scoreImagePage(request):
    return render(request, 'scorepage.html')

def predictImage(request):
    fileObj = request.FILES['filePath']
    fs = FileSystemStorage()
    filePathName = fs.save('images/' + fileObj.name, fileObj)
    filePathName = fs.url(filePathName)
    modelName = request.POST.get('modelName')
    scorePrediction, img_uri = predictImageData(modelName, '.' + filePathName)
    context = {'scorePrediction': scorePrediction, 'filePathName': filePathName,
'img_uri': img_uri}
    return render(request, 'scorepage.html', context)

def predictImageData(modelName, filePath):
    img = Image.open(filePath).convert("RGB")
    resized_img = img.resize((32, 32), Image.ANTIALIAS)
    img_uri = to_data_uri(resized_img)
    input_image = Image.open(filePath)
    preprocess = transforms.Compose([
        transforms.Resize(32),
        transforms.CenterCrop(32),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224,
0.225]),
    ])
    input_tensor = preprocess(input_image)
    input_batch = input_tensor.unsqueeze(0)

    sess = onnxruntime.InferenceSession(
r'C:\PycharmProjects\MPPR_DZ3\media\models\cifar100.onnx') # <-Здесь
требуется указать свой путь к модели
    outputOFModel = np.argmax(sess.run(None, {'input': to_numpy(input_batch)}))
    score = imageClassList[outputOFModel]

    return score, img_uri

def to_numpy(tensor):
    return tensor.detach().cpu().numpy() if tensor.requires_grad else
tensor.cpu().numpy()

def to_image(numpy_img):
```

```

img = Image.fromarray(numpy_img, 'RG')
return img

def to_data_uri(pil_img):
    data = BytesIO()
    pil_img.save(data, "JPEG") # pick your format

    data64 = base64.b64encode(data.getvalue())
    return u'data:image/jpeg;base64,' + data64.decode('utf-8')

```

Файл urls.py:

```

from django.contrib import admin
from django.urls import path
from django.conf.urls.static import static
from django.conf import settings
from . import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.scoreImagePage, name='scoreImagePage'),
    path('predictImage', views.predictImage, name='predictImage'),
]

urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

```

Файл scorepage.html:

```

<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="windows-1251">
<title>DZ1</title>
<style>
/* Add some padding on document's body to prevent the content
to go underneath the header and footer */ body{
    padding-top: 60px;
    padding-bottom: 40px;
}

.fixed-header, .fixed-footer{
    width: 100%;
    position: fixed;
    background: #333;
    padding: 10px 0;
    color: #fff;
}

.fixed-header{
    top: 0;
}

.fixed-footer{
    bottom: 0;
}

.container{
    width: 80%;
    margin: 0 auto; /* Center the DIV horizontally */
}

nav a{
    color: #fff;
    text-decoration: none;
    padding: 7px 25px;
}

```

```

    display: inline-block;
  }
</style>
</head>
<body>
  <div class="fixed-header">
    <div class="container">

</div> </div> <div class="container">
<form action="predictImage" method="post" enctype="multipart/form-data">
  {% csrf_token %}

      <div class="col-md-4 col-sm-4">
<label for="FilePath">Select:</label>
</div> <input name="filePath" type="file"><br><br>
<input type="submit" value="Submit" >
</form> </div> <div> <br>  {% if scorePrediction %}
      <h3>The classification is : {{scorePrediction}}</h3>
  {% endif %}

  </div>

<div>  {% if scorePrediction %}
      
  {% endif %}
  </div>
<div class="fixed-footer">
<div class="container"></div>
</div></body>
</html>

```

Работа приложения:

Select:

Выберите файл

Файл не выбран

Submit

The classification is : bear



Select:

Выберите файл

Файл не выбран

Submit

The classification is : bear





Select:

Выберите файл Файл не выбран

Submit

**The classification is : forest**



Select:

Выберите файл Файл не выбран

Submit

**The classification is : forest**



Select:

Выберите файл Файл не выбран

Submit

**The classification is : porcupine**



Select:

Выберите файл Файл не выбран

Submit

**The classification is : porcupine**

