

ФАЙЛ MAIN.PY

```
model = torchvision.models.detection.fasterrcnn_resnet50_fpn(pretrained=True)
```

- **Описание:** Эта строка кода создаёт объект `model`, который представляет собой предобученную модель Faster R-CNN с ResNet-50 и Feature Pyramid Network (FPN) в качестве базовой сети.
- **Назначение:** Модель предназначена для задач детекции объектов, где она не только классифицирует объекты на изображении, но и определяет их местоположение с помощью ограничивающих рамок. **Как это работает**

- **Извлечение количества входных признаков:**

- Строка кода извлекает значение атрибута `in_features` из слоя `cls_score`.
- Это значение используется, чтобы знать, сколько признаков передается на вход классификационного слоя.

- **Зачем это нужно:**

- Это полезно, когда необходимо заменить или модифицировать классификационный слой для адаптации модели к новым задачам. Например, если вы хотите дообучить модель на своем наборе данных с другим числом классов, вам нужно создать новый классификационный слой с правильным числом входных признаков.

```
in_features = model.roi_heads.box_predictor.cls_score.in_features
```

- **Описание:** Эта строка кода извлекает значение атрибута `in_features` из слоя `cls_score` в компоненте `box_predictor` модели Faster R-CNN.
- **Назначение:** Значение `in_features` указывает на количество входных признаков, которые принимает на вход слой `cls_score` для предсказания классов объектов.
- **Использование:**
 - Это значение важно, когда нужно заменить существующий классификационный слой новым. Например, для адаптации модели к новому набору данных с другим числом классов необходимо знать размерность входных данных для нового слоя.

```
model.roi_heads.box_predictor = FastRCNNPredictor(in_features, num_classes)
```

- **Замена классификационного слоя:**

- Строка кода заменяет существующий модуль `box_predictor` в компоненте `roi_heads` модели Faster R-CNN новым модулем `FastRCNNPredictor`.
- Новый `FastRCNNPredictor` инициализируется с указанным количеством входных признаков (`in_features`) и количеством классов (`num_classes`).

- **Зачем это нужно:**

- Предобученные модели, такие как Faster R-CNN, часто обучаются на общих наборах данных, таких как COCO, с фиксированным количеством классов. Если у вас есть другой набор данных с разными классами, вам нужно адаптировать модель.
- Замена `box_predictor` позволяет изменить количество классов, которые может предсказывать модель, что делает ее пригодной для новой задачи.

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

- **Описание:** Эта строка кода создает объект `torch.device`, который указывает, будет ли PyTorch использовать GPU ('cuda') или CPU ('cpu') для выполнения вычислений, и присваивает его переменной `device`.
- **Назначение:** Автоматически определяет, использовать ли GPU или CPU для вычислений, и устанавливает соответствующее устройство.
- **Использование:** Переменная `device` используется для переноса моделей и данных на выбранное устройство, что необходимо для выполнения операций в PyTorch.