**Goal 1: Creating a Pipeline for Object Detection**

**Introduction**

Object detection is a fundamental task in computer vision, essential for applications like surveillance, autonomous driving, and healthcare. This report presents a pipeline for training, testing, and inference using YOLOv5 and Faster R-CNN on the Mask Wearing Dataset.

**Dataset**

The Mask Wearing Dataset, available at [Roboflow](https://public.roboflow.com/object-detection/mask-wearing), consists of labeled images distinguishing masked and unmasked faces. The dataset is preprocessed and split into training, validation, and test sets.

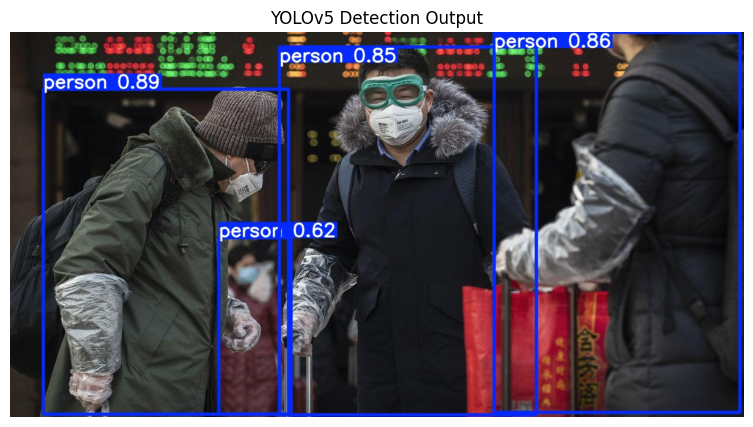
**Model 1: YOLOv5**

**Training Pipeline**

The training process involves installing dependencies, preparing the dataset in YOLO format, and training the model using a predefined configuration. Performance is monitored using tools like TensorBoard or Weights & Biases.

**Testing and Inference**

Model evaluation is conducted using validation datasets, and inference is performed on test images to detect mask-wearing patterns.



**Model 2: Faster R-CNN**

**Training Pipeline**

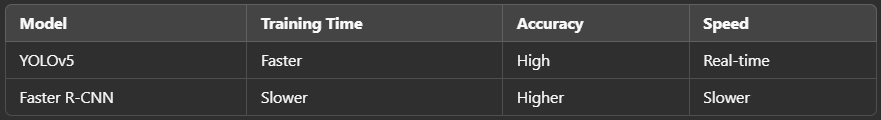
The Faster R-CNN model is trained using TensorFlow/Keras with a pre-trained backbone. The dataset is converted into an appropriate format, and training is performed using standard loss functions and optimizers.

**Testing and Inference**

Performance evaluation is done using test datasets, and the model is used to make predictions on unseen images.



**Comparison of Models**

****

**Goal 2: CNN as a Feature Extractor**

**Introduction**

Feature extraction using Convolutional Neural Networks (CNNs) helps visualize how different layers detect patterns in images. This experiment extracts features at three different layers from a pre-trained CNN model and displays them.

**Dataset**

The Stanford Car Dataset ([link](http://ai.stanford.edu/~jkrause/cars/car_dataset.html)) contains images of various car models, suitable for feature extraction tasks.

**Model Selection**

The following pre-trained CNN models were used:

* ResNet-50
* VGG-16
* InceptionNetV3

**Implementation Steps**

A pre-trained model is loaded, and three layers are selected for feature extraction. Features are then processed and visualized to analyze how CNNs detect edges, textures, and objects at different depths.

**Results**

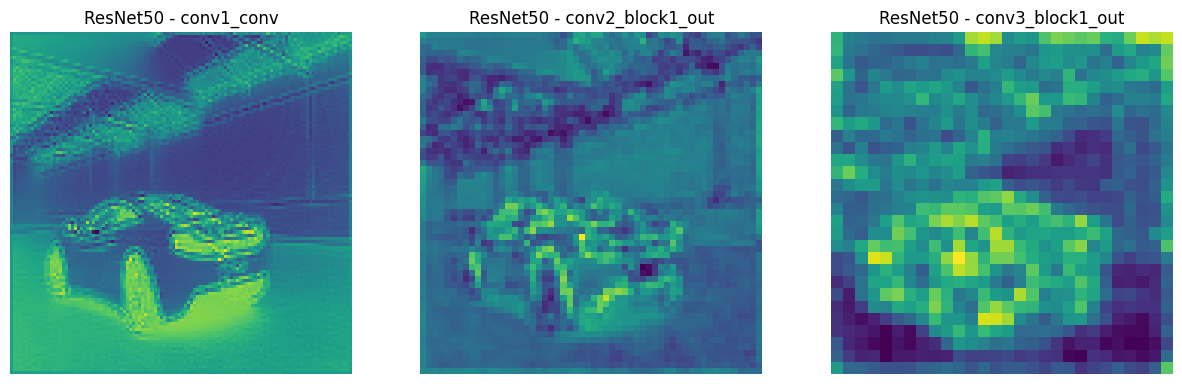
Feature maps at different layers provide insight into how CNNs detect edges, textures, and objects. Lower layers capture simple features like edges, while deeper layers learn complex patterns.

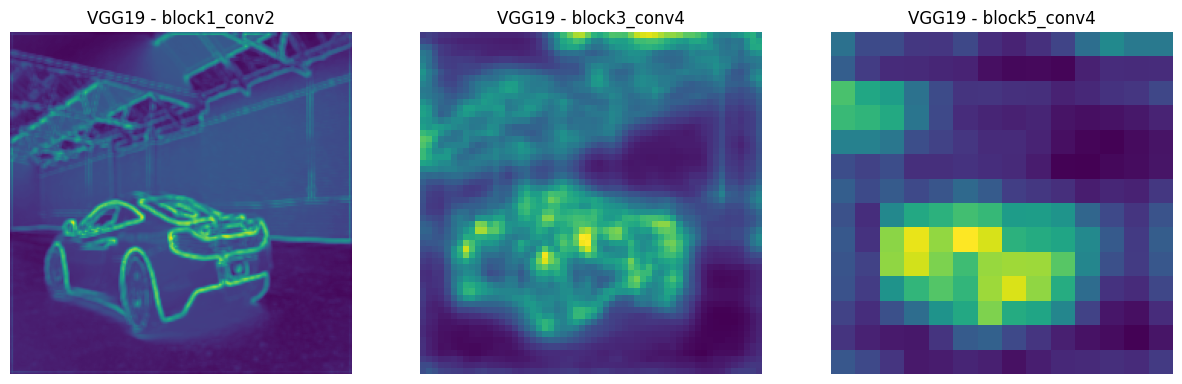
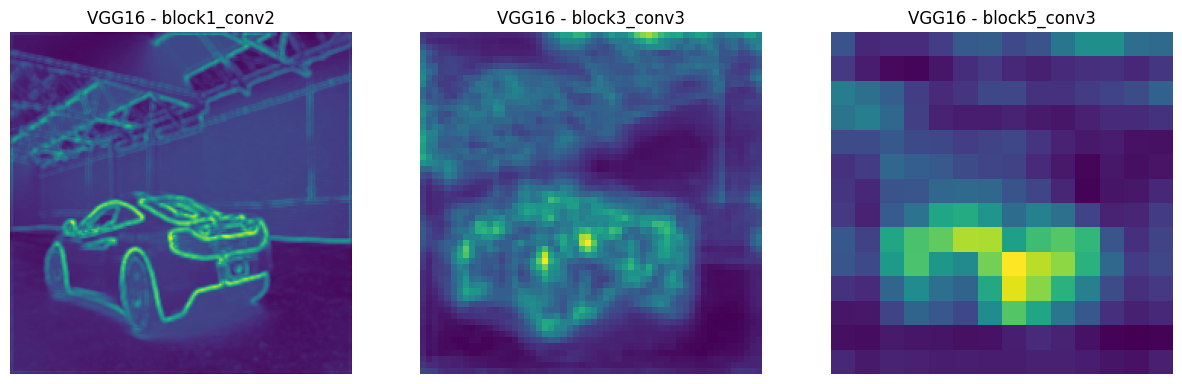
**Conclusion**

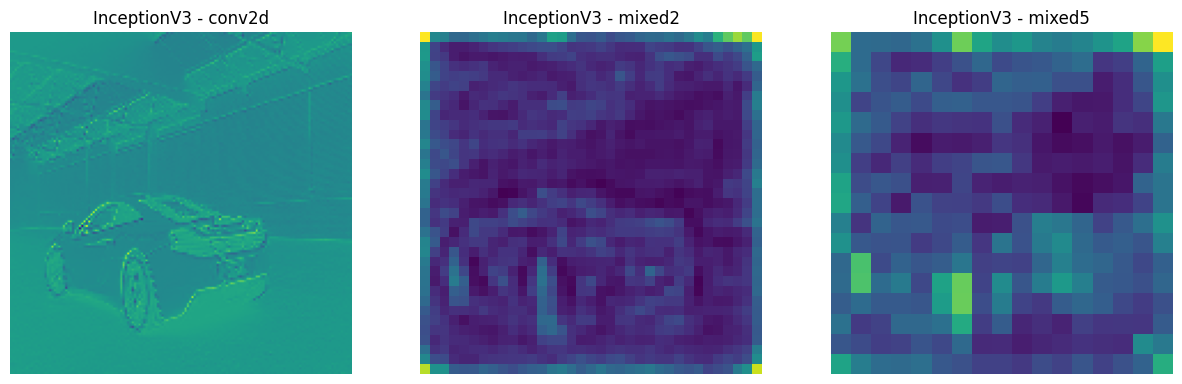
* The YOLOv5 model provides fast and efficient object detection for mask-wearing classification, whereas Faster R-CNN achieves higher accuracy but at a computational cost.
* Feature extraction using CNNs enables visualization of hierarchical feature learning, demonstrating the model’s ability to capture different image properties at various depths.

**5 randomly chosen images**

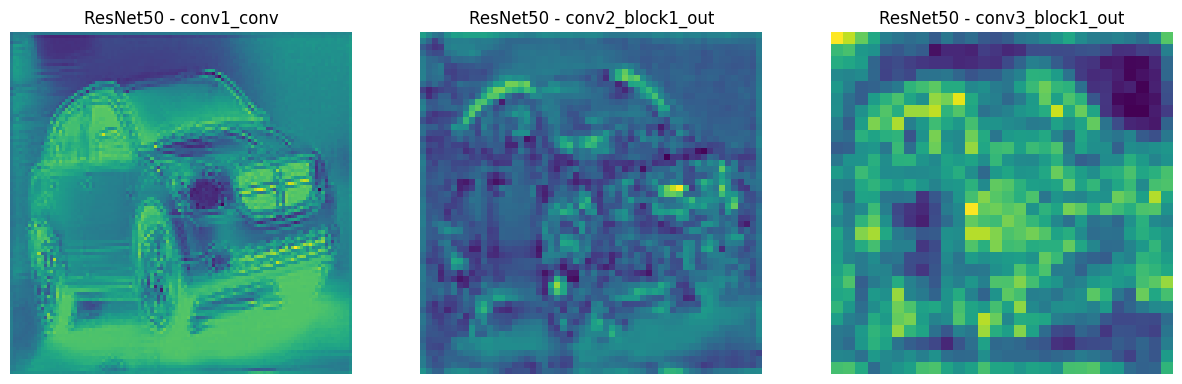
1.

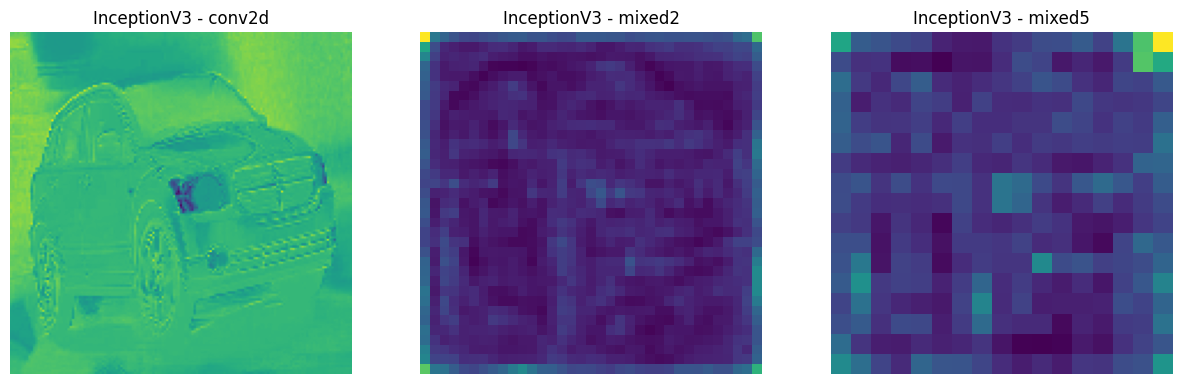
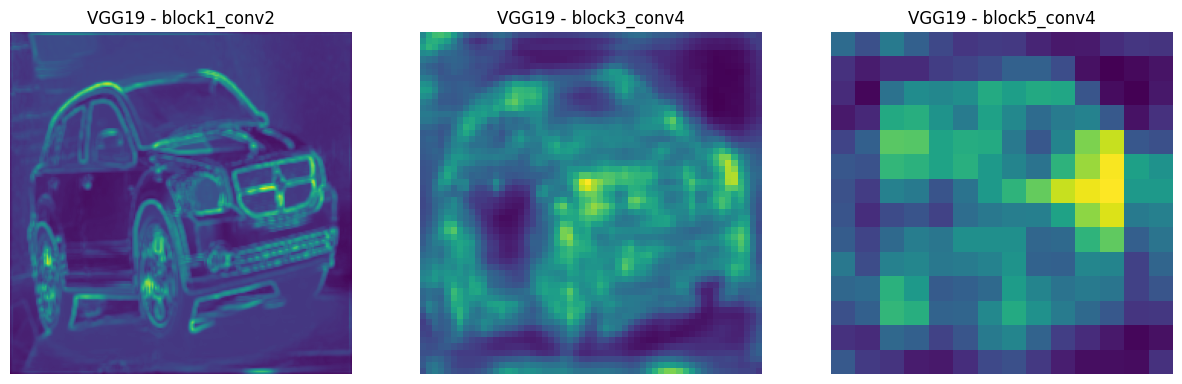
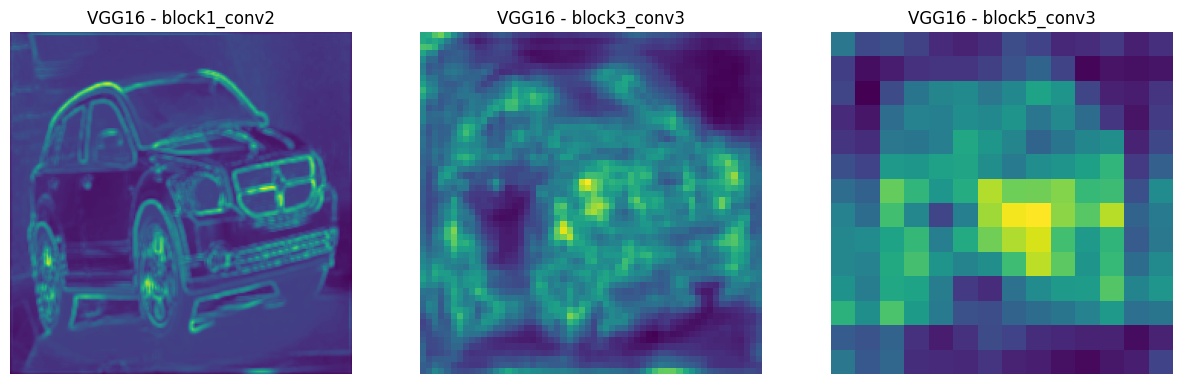




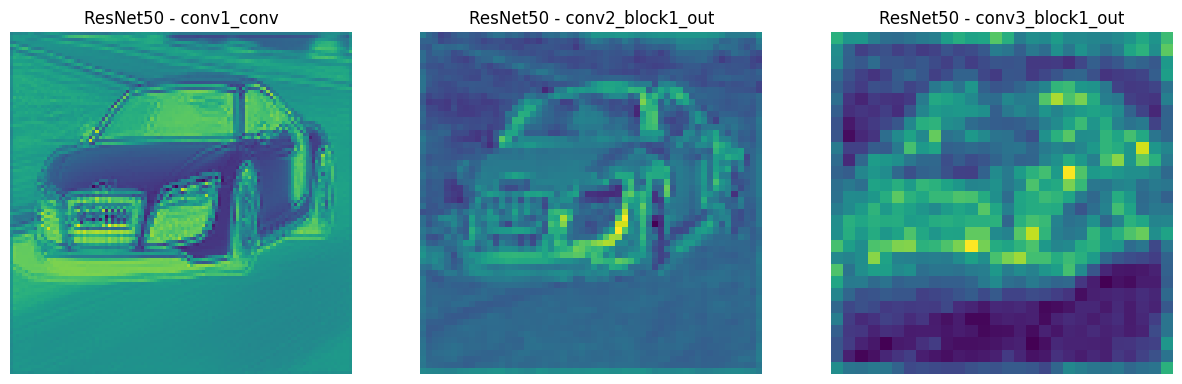


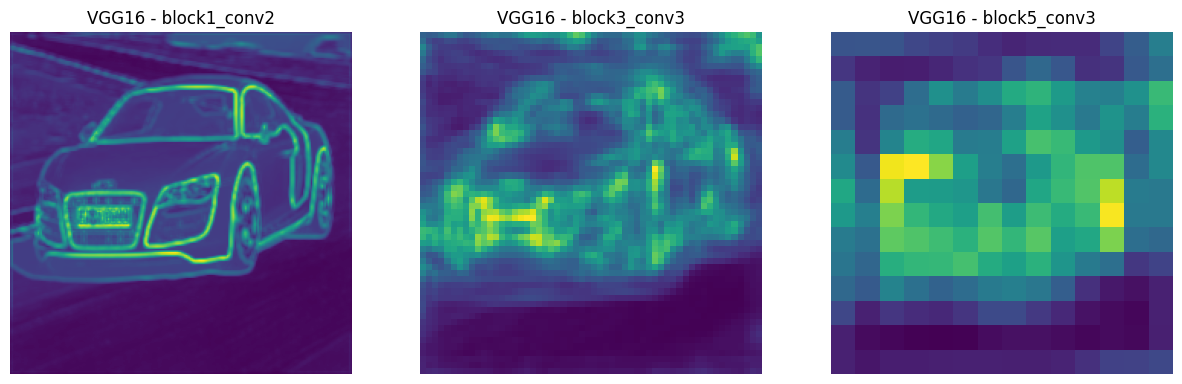
2.

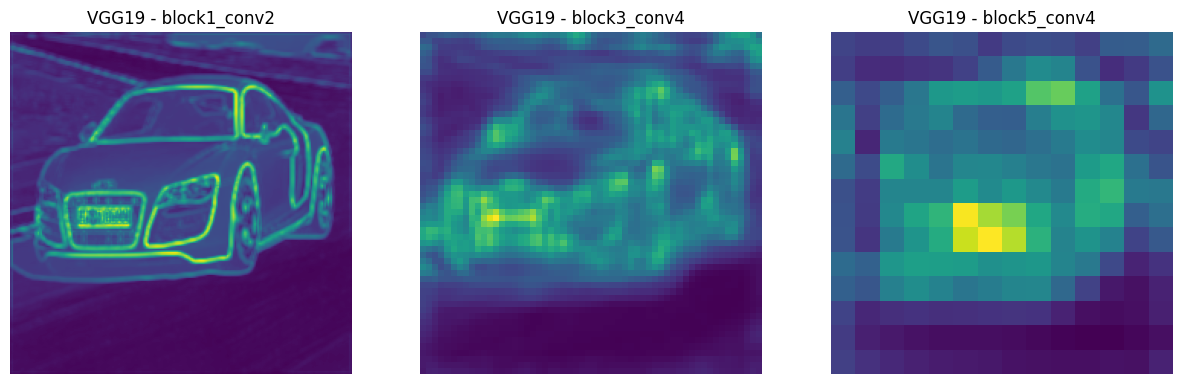


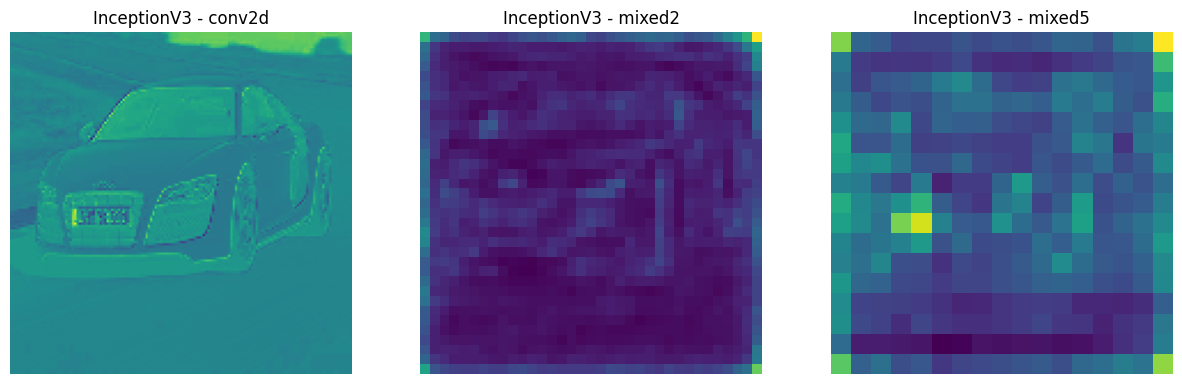


3.

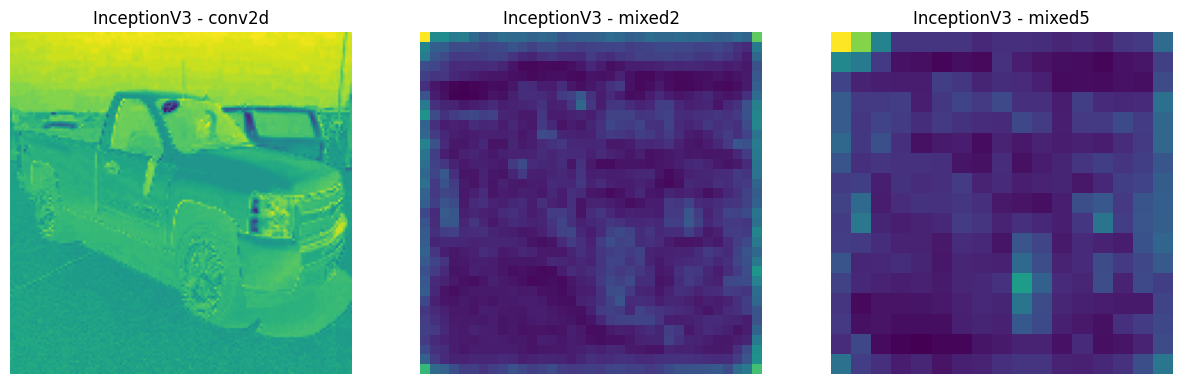
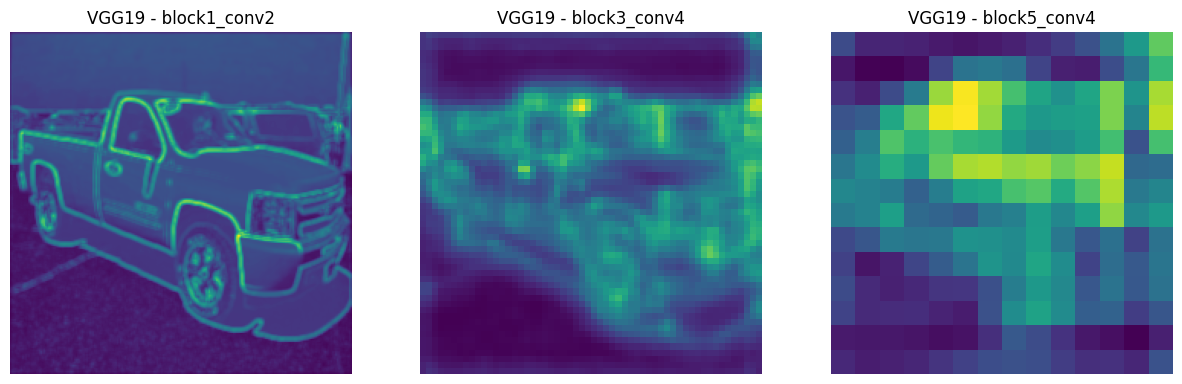
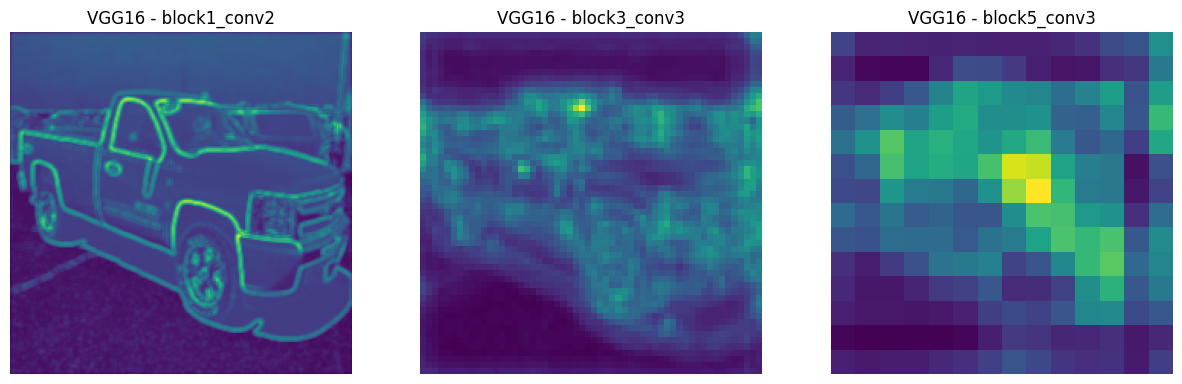
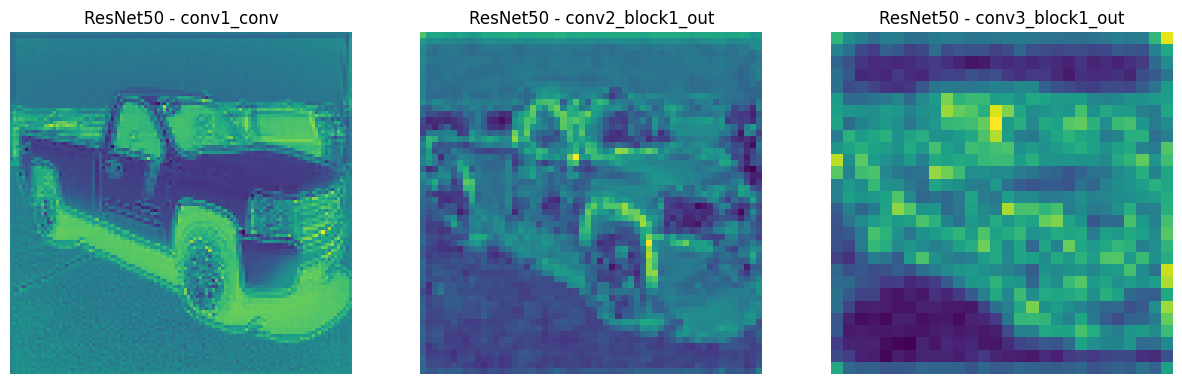








4.



5.

