Part 1: Documentation of Architectures

1. VGG19 (Visual Geometry Group)

Step-by-Step Architecture:

1. **Input:** Takes an image input of shape (224, 224, 3).
2. **Convolutional Blocks:** It consists of 5 blocks of convolutional layers.
   - **Block 1 & 2:** Two 3x3 Convolutional layers followed by Max Pooling.
   - **Block 3, 4, & 5:** Four 3x3 Convolutional layers followed by Max Pooling.
3. **Feature Maps:** The filters double in depth at every block (64 $\to$ 128 $\to$ 256 $\to$ 512).
4. **Classification Head:** traditionally includes three Fully Connected (Dense) layers (4096, 4096, 1000). In your notebook, this was modified to a flattened layer followed by Dense(512), Dropout, and the final Softmax layer.

   Diagram Concept:

   Imagine a straight pipe where the image shrinks in width/height but grows in depth (number of filters) as it passes through 16 convolutional layers and 3 dense layers.

Reference:

- Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv preprint arXiv:1409.1556.

---

2. Inception V1 (GoogLeNet)

Step-by-Step Architecture:

1. **Stem:** Basic convolution and pooling layers to reduce input resolution quickly.
2. **Inception Modules:** The core innovation. Instead of choosing a filter size (1x1, 3x3, or 5x5), this block performs *all* of them in parallel and concatenates the outputs.
3. **1x1 Convolutions:** Used inside modules to reduce dimensionality (number of channels) before expensive 3x3 and 5x5 convolutions, saving computational cost.
4. **Auxiliary Classifiers:** Side branches used during training to inject gradients earlier in the network (solving the vanishing gradient problem), though these are usually removed for inference.

   Diagram Concept:

   The network looks "wide" rather than just deep. Data splits into four parallel paths within a block and merges back together.

Reference:

- Szegedy, C., et al. (2015). *Going Deeper with Convolutions*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

---

3. ResNet50 (Residual Network)

Step-by-Step Architecture:

1. **Residual Learning:** addresses the "vanishing gradient" problem in deep networks.
2. **Skip Connections:** The input to a block x is added to the output of the block $F(x)$, result is $F(x) + x$. This allows the network to learn "residuals" (changes) rather than a full mapping.
3. **Bottleneck Blocks:** Uses 1x1 convolutions to squeeze dimensions, a 3x3 convolution to process, and a 1x1 to expand back, improving efficiency.
4. **Global Average Pooling:** Replaces the heavy fully connected layers found in VGG, feeding directly into the final classification layer.

Diagram Concept:

A deep stack of layers where arrows curve around blocks, allowing information to "skip" layers, facilitating the training of very deep networks (50 layers).

Reference:

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. Proceedings of the IEEE CVPR.

---

4. MobileNetV2

Step-by-Step Architecture:

1. **Depthwise Separable Convolutions:** Splits standard convolution into two parts:
   - *Depthwise:* Filters apply to a single input channel.
   - *Pointwise:* 1x1 convolution combines the outputs. This drastically reduces parameters.
2. **Inverted Residuals:** Opposite of ResNet. The network expands dimensions to high depth, performs convolution, and then squeezes back to low dimensions (bottleneck).
3. **Linear Bottlenecks:** Removes non-linearities (ReLU) in the narrow layers to preserve information.

Diagram Concept:

A streamlined architecture designed for mobile devices, characterized by thin bottleneck layers connecting thick intermediate expansion layers.

Reference:

- Sandler, M., et al. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. Proceedings of the IEEE CVPR.

Part 2: Comparative Analysis of Models

Experimental Results Summary

Based on the code outputs from your files, here is the performance comparison on the EuroSAT dataset:

Metric	**ResNet50**	**Inception V1**	**MobileNetV2**	**VGG19 (Custom)**
Test Accuracy	96.15%	94.69%	91.83%	~11 - 30%*
Precision	96.11%	94.61%	91.61%	Low
Recall	95.84%	94.36%	91.19%	Low
F1-Score	95.93%	94.41%	91.30%	Low

**Note: The VGG19 model in `Untitled_1` was trained from scratch on a reduced dataset (5000 images) and failed to converge properly compared to the Transfer Learning models used for the others.*

Pros and Cons

Architecture	**Pros**	**Cons**
ResNet50	**Highest Accuracy.** Skip connections allow it to be very deep without training errors. Excellent feature extraction for complex textures (like satellite land use).	Computationally expensive compared to MobileNet. High memory usage.
Inception V1	High accuracy. Efficient use of computing resources via 1x1 convolutions. Handles multi-scale features well (captures both fine and coarse details).	Complex architecture design. Harder to implement manually or modify.

Architecture	Pros	Cons
MobileNetV2	**Fastest and lightest.** Extremely low parameter count. Perfect for deployment on phones or edge devices (drones/satellites).	Slightly lower accuracy than the heavy hitters (ResNet/Inception). Struggles with very fine-grained distinct features compared to deeper networks.
VGG19	Very simple architecture (uniform 3x3 blocks). Good for understanding basic CNN concepts.	**Extremely heavy** (huge number of parameters). Slow to train. Prone to overfitting on small datasets without pre-training.

Conclusion: The Best Model

ResNet50 is the best performing architecture for this specific task (EuroSAT Land Use Classification) with an accuracy of **96.15%**.

Why it performed better:

1. **Transfer Learning:** You utilized weights pre-trained on ImageNet. Satellite images, while distinct, share low-level features (edges, textures, shapes) with ImageNet data. ResNet's deeper architecture allows it to learn higher-level abstractions of these features more effectively than MobileNet.
2. **Residual Connections:** The EuroSAT dataset can have subtle differences between classes (e.g., *Forest* vs. *Pasture*). The depth of ResNet allow it to learn these nuances, while the residual connections ensure the gradient signal remains strong during training.
3. **VGG Failure:** The VGG model failed because it was trained **from scratch** on a small subset of data. Deep CNNs require massive amounts of data to learn filters from scratch. The other three models succeeded because they leveraged Transfer Learning.