Differences and Effects of Increasing Model Width and Depth in Deep Learning

1. Increasing Width

- Adds more neurons to each other
- Improves features extraction but increases computational cost
- Increases the risk of overfitting

2. Increasing Depth:

- Adds more layers to the model
- learns of more complex patterns but may cause gradient vanishing
- Increases model expressiveness and makes training more challenging

Problems and Solutions When Increasing Model Depth

1. Methods to Solve Gradient Vanishing
    - Residual Connections: Skip connections in ResNet allow gradients to flow directly to earlier connections
    - Batch Normalizations: Stabilizes gradient flow by normalizing layer inputs
    - Appropriate Activation Functions: Using ReLU or LeakyReLU

2. EfficientNet's Compound
    - Balances width, depth and resolution simultaneously
    - Uses a single coefficient

$$depth: d = \alpha^{\phi}$$

$$width: w = \beta^{\phi}$$

$$resolution: r = \gamma^{\phi}$$

$$\alpha \cdot \beta^2 \cdot \gamma^2 \simeq 2 \qquad (\alpha \geq 1, \beta \geq 1, \gamma \geq 1)$$

3. Key Architectural Design Elements
   - Attention Mechanism: Allows the model to focus on important parts of the input
   - Bottleneck Architecture: Improves computational efficiency while effectively performing feature extraction
   - Convolution Kernel Size: Determines the size of the receptive field
   - Activation Function Selection: Adds non-linearity and enhance expressive power.
   - Normalization Techniques: Improves training stability and generalization performance.