

Paper 1

- **Paper Name:** *Classification of Brain Tumors on MRI Images Using Convolutional Neural Network Model EfficientNet.*
- **Paper Link:** <http://jurnal.iaii.or.id>.
- **Preprocessing Steps:**
 - **Data Augmentation:** Added to the dataset to manipulate data and improve results.
 - **Normalization:** Implemented min-max normalization.
 - **Input Resizing:** Images were resized to a constant value of **128 x 128** pixels.
- **Model Name and Properties:**
 - **Models used:** EfficientNetB0 and EfficientNetB7.
 - **Architecture Layers:** Input layer, GlobalAverage Pooling2D layer, Dropout layers, and Dense layers.
- **Hyperparameters:**
 - **Epochs:** 100.
 - **Learning Rate:** 0.00146.
 - **Optimizer:** Adam (Scenario 1) and RMSprop (Scenario 2).
 - **Dropout Value:** 0.2.
 - **Dense Layer Units:** 1024 and 128.
 - **Activation Functions:** Relu (for hidden dense layers) and Softmax (for the output layer).
 - **Loss Function:** Categorical Crossentropy.

Paper 2

- **Paper Name:** An Integrated Deep Learning Model with EfficientNet and ResNet for Accurate Multi-Class Skin Disease Classification.
- **Paper Link:** <https://doi.org/10.3390/diagnostics15050551>.
- **Preprocessing Steps:**
 - **Dataset Split:** Data was distributed into Training (**80%**), Validation (**10%**), and Testing (**10%**).
 - **Normalization:** Features were normalized using **Batch Normalization** to stabilize training.
 - **Zero Padding:** Applied at the initial architecture stage of ResNet50 for spatial dimension protection.
 - **Resizing:** Standardized to fit the requirements of the pre-trained models (typically 224x224 for ResNet/EfficientNet, though specific target values for this paper are managed by the "compound scaling" of EfficientNet).
- **Model Name and Properties:**
 - **Model:** A fusion-based deep learning model (ensemble).
 - **Merged Architectures:** EfficientNet-B0, EfficientNet-B2, and ResNet50.
 - **Fusion Mechanism:** Features from the three branches are concatenated and

transmitted to dense and dropout layers.

- **Hyperparameters:**
 - **Regularization:** Uses **Dropout** and **Batch Normalization** to reduce overfitting.
 - **Activation Functions:** **SWISH** (used in EfficientNet blocks) and **ReLU** (used in ResNet stages).
 - **Pooling:** Average Pooling and Max Pooling are utilized to decrease dimensions.
 - **Classification Layer:** Final layer uses a fully connected layer for multi-class predictions.