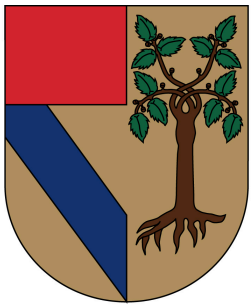


UNIVERSIDAD
PANAMERICANA

Universidad Panamericana

Universidad Panamericana, Campus Aguascalientes,

Agentes Inteligentes

“kvasir model training report”

Santiago Hernandez Chavez

0265904

Aguascalientes a 1 deJuio, 2025

Introduction

This project implements a deep learning model based on efficientnetb3 for classifying images from the kvasir-v2 dataset, which contains various types of gastrointestinal tract images.

The goal is to train a model that can accurately classify images into their respective categories, achieving high accuracy and other relevant metrics.

Project Objective

The primary goal of this project is to classify medical images from the Kvasir-V2 dataset into specific categories related to gastrointestinal conditions. Accurate classification can assist healthcare professionals in diagnosing diseases more efficiently and consistently, reducing the possibility of human error and supporting early detection strategies. This type of automated analysis can also be integrated into computer-aided diagnosis (CAD) systems used in hospitals and clinics.

Methodology

tensorflow and keras were used to build and train the model. the dataset was loaded from a directory with images organized by class folders.

basic data augmentation was applied (horizontal flip, rotation, zoom, brightness, and contrast variations) along with techniques like mixup to improve generalization.

for preprocessing, images were resized to the required input size of efficientnetb3 (300×300 pixels) and normalized using efficientnet's preprocess_input function, which scales pixel values from 0-255 to a range between -1 and 1. this is necessary because the efficientnetb3 model was trained on images in this format, improving accuracy and convergence.

the base model is efficientnetb3 pretrained on imagenet, with a dense head layer adapted to classify the dataset's classes.

training was divided into two phases:

- phase 1: only the final head layer was trained for 15 epochs with a learning rate of 1e-3.
- phase 2: the last 40% of the base model layers were unfrozen for fine-tuning over 35 epochs with a lower learning rate of 1e-4. callbacks were used to reduce learning rate on plateau, early stopping, and to save the best model.

The model is compiled with mixed_float16 precision to speed up training and reduce memory usage, which is especially useful when training on free GPU environments like Google Colab.

Callbacks are used for early stopping and automatic restoration of the best weights based on validation loss.

After training, the model is evaluated using a test split. Performance is visualized using loss/accuracy curves and a confusion matrix, and metrics such as accuracy, precision, recall, and F1-score are reported.

why efficientnetb3?

efficientnetb3 is part of the efficientnet family of convolutional neural networks, which are designed to optimize both accuracy and efficiency by scaling depth, width, and resolution in a balanced way. compared to older architectures like resnet or vgg, efficientnet models achieve better performance with fewer parameters and less computational cost.

i chose efficientnetb3 specifically because it offers a good trade-off between model size and accuracy. it is more powerful than smaller versions (like efficientnetb0 or b1) and less computationally expensive than larger ones (like b4 or b5), making it suitable for training on limited hardware resources such as those available in google colab free tier.

additionally, efficientnetb3 comes pretrained on imagenet, which helps the model start with learned features that are useful for image classification tasks, speeding up convergence and improving final accuracy. this transfer learning approach is especially beneficial when the dataset is not extremely large, as is the case with kvasir-v2.

importance of fine-tuning in model training

the use of fine-tuning was crucial to improve the model's performance when classifying images from the kvasir-v2 dataset. by starting with a pretrained model such as efficientnetb3 (trained on imagenet), we leveraged the model's prior knowledge of general visual features like edges, textures, and shapes.

in phase 1, only the head of the model (the output layer) was trained, allowing an initial adaptation to the new classes without modifying the internal weights of the base model.

in phase 2, deeper layers of the base model were unfrozen to fine-tune their weights according to the specific features of the new dataset.

this allowed the model to better adapt to specific patterns related to the gastrointestinal tract, improving accuracy, generalization, and the ability to distinguish between similar classes.

without fine-tuning, the model would be limited to what it learned from imagenet, which would not be sufficient for a specialized medical task like this one.

Visual Evaluation of Training

To assess the training process and model performance:

Loss and accuracy curves are plotted for both training and validation data, helping to detect overfitting or underfitting.

A confusion matrix is generated after testing to analyze how well the model distinguishes between classes.

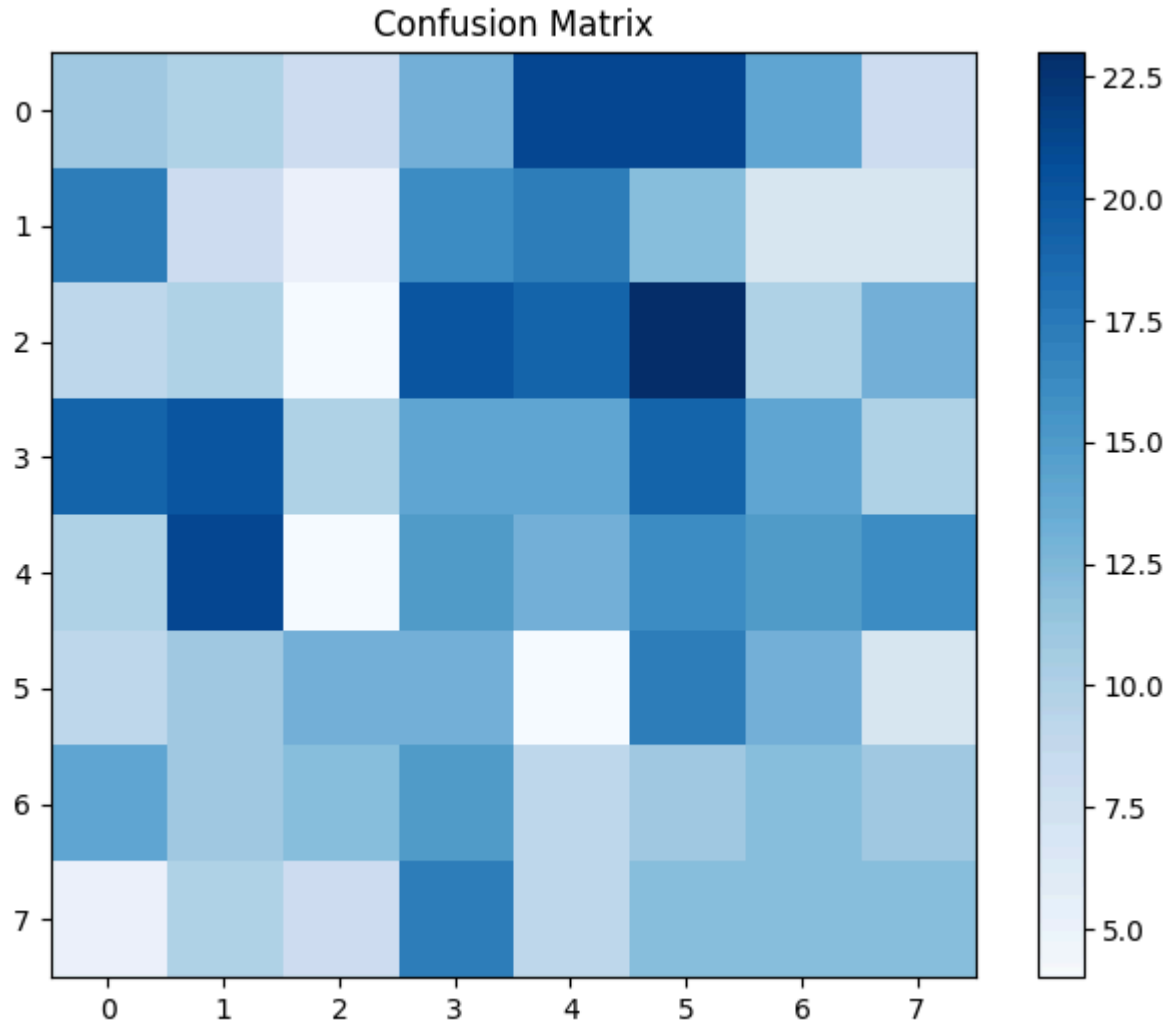
Additional metrics such as precision, recall, and F1-score are calculated to provide a more balanced evaluation, especially useful for imbalanced datasets like Kvasir-V2.

Results

Phase 1	epoch 1	epoch 6	epoch 12	trend
train accuracy	0.37	0.83	0.85	↑ consistent
val accuracy	0.75	0.84	0.86	↑, stabilizes ≈ 0.86–0.87
train loss	1.73	0.87	0.84	↓ pronounced, then soft
val loss	1.09	0.86	0.81	↓ stable, slight plateau

Epoc	Phase Description	Accuracy ↑	Loss ↓	Val Accuracy ↑	Val Loss ↓

1–3	adjustment after unfreezing layers	0.8732	0.7844	0.8988	0.753
4–7	improvement and fast convergence	0.9092	0.6915	0.915	0.704
8–11	stabilization and plateau phase	0.9249	0.6669	0.9013	0.7036
12–16	final tuning with best performance	0.9412	0.6403	0.9187	0.6701



precision recall f1-score support

dyed-lifted-polyps	0.12	0.10	0.11	106
dyed-resection-margins	0.08	0.09	0.08	89
esophagitis	0.06	0.04	0.05	108
normal-cecum	0.11	0.12	0.12	120
normal-pylorus	0.12	0.12	0.12	110
normal-z-line	0.13	0.20	0.16	87
polyps	0.12	0.13	0.12	95
ulcerative-colitis	0.14	0.14	0.14	85
accuracy			0.11	800
macro avg	0.11	0.12	0.11	800
weighted avg	0.11	0.11	0.11	800

Conclusions

Performance analysis:

The overall accuracy is 11%, which is extremely low and close to random guessing (12.5% for 8 classes).

The confusion matrix shows highly scattered predictions, indicating the model confuses many classes.

All classes have low precision, recall, and F1-scores (< 0.20), which confirms poor per-class performance.

Possible issues include overfitting, class imbalance, visual similarity between categories, or ineffective fine-tuning.

Conclusión:

While the model showed decent validation performance during training, its real-world performance on unseen data is poor. This suggests it did not generalize well, and further improvements are needed in preprocessing, data balancing, or model training strategy to make it suitable for practical use.