**Assignment 1**

Data Augmentation

Tutor: Bram Boerenkamp

**Assignment Overview**

**Problem statement**

This assignment focuses on the importance of data augmentation in machine learning, particularly when working with small or limited datasets. Data augmentation is a technique that artificially increases the diversity of a training set by applying transformations to images, allowing models to generalize better and avoid overfitting. This is especially useful in fields like Biomedical Engineering, where datasets can be limited and costly to obtain. By learning data augmentation techniques, you'll gain skills to improve model robustness and handle scenarios with restricted data, making your models more adaptable and capable of accurate predictions in real-world applications.

**The assignment**

In this assignment, each of you will receive a unique 10% subset of the CIFAR-10 dataset, a collection of 32x32 color images that represent 10 object categories (e.g., airplanes, automobiles, birds). By working on individual subsets, you’ll simulate a limited-data scenario, giving you the opportunity to explore how data augmentation can enhance a model’s performance with constrained data. Through this assignment, you will gain hands-on experience in implementing augmentations, training a model, and evaluating its performance, an essential skill for fields with limited data availability, like Biomedical Engineering.

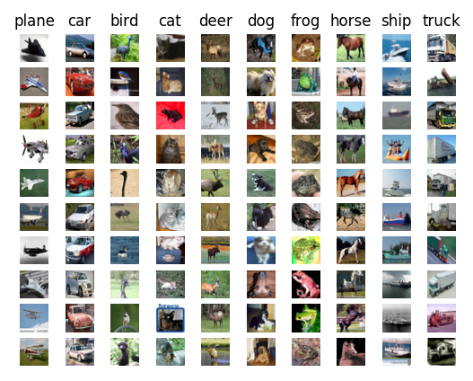


Figure 1: CIFAR-10 Image clasification dataset

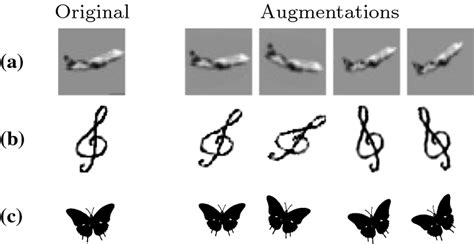


Figure 2: Examples of data augmentation applied to data of **CIFAR (a),** HOMUS (b) and MPEG-7 (c) datasets. The generated samples might depict several deformations at the same time, namely rotation, zoom and translation.

**Assignment Instructions**

1. **Load and Explore the Dataset**

* Run the code to load a 10% subset of the CIFAR-10 dataset. Each of you will get a unique subset. Review the data carefully to understand the classes and visuals you'll work with. Then, visualize some sample images to get a feel for potential augmentations.

1. **Train the Baseline Model**

* Train the given CNN model on your reduced dataset *without any augmentation*. Record the accuracy, precision, and recall—these metrics will serve as a baseline to compare against once augmentation is applied.

1. **Apply Data Augmentation**

* Use the code cell provided to apply your specific data augmentation method. Within your tutor group each of you will choose a different data augmentation method. The available data augmentation methods are:
  + Rotations, flipping, scaling, or translation. (or other geometric augmentations)
  + Gaussian noise, salt-and-pepper noise, or speckle noise. (or other noise augmentations)
  + Brightness adjustments, contrast adjustments, or color jittering. (or other color augmentations)

For each of these categories you will have to pick some methods based on literature or try them all and find out which works best by trial and error. This is all up to you but remember that the goal it to enhance the model performance as much as possible. (tip: you can also look up the model architecture and find out if some data augmentation methods work better than others with this model)

* You can use libraries like tf.image (TensorFlow), ImageDataGenerator (Keras) or any other library that suits you needs but remember to make sure that you always understand the underlying mechanisms by printing augmented images and/or reading the libraries documentation.
* !! Remember to apply augmentation only to the training set !!

1. **Train the Model with Augmentation**

* Now train your CNN model again using the augmented dataset. Track your training and validation metrics (accuracy, precision, recall) to see if augmentation improves performance.

1. **Evaluate and Compare Results**

* Calculate and document the final test accuracy, precision, and recall. Summarize these metrics to compare the baseline (no augmentation) and augmented models. Write a brief analysis on the impact of each augmentation technique and your overall findings.

1. **Wrap-Up**

* Reflect on the assignment: How did augmentation affect model performance? Which techniques seemed most effective? Summarize key insights and consider how augmentation might help in real-world biomedical engineering tasks with small or imbalanced datasets.

1. **LLM Usage Report**

* If you used a language model (e.g., ChatGPT, Copilot) for this assignment, complete a reflection using the provided template for LLM usage. You can find this template on canvas. Remember that we will check on LLM usage and that it will be considered **fraud** when you have used a LLM without reporting that you did. The template on LLM usage is there for you to learn from your LLM usage experience but will not count to the final grade of this assignment.

**Useful Resources**

These resources will help you get started with your assignment:

* [Jupyter notebook tutorial (for refreshment if necessary)](https://www.dataquest.io/blog/jupyter-notebook-tutorial/)
* [Data augementation theory](https://towardsdatascience.com/complete-guide-to-data-augmentation-for-computer-vision-1abe4063ad07)
* [keras image preprocessing documentation](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image)
* [Tensorflow data augmentation tutorial](https://www.tensorflow.org/tutorials/images/data_augmentation" \l "custom_data_augmentation)

**What to submit**

* The filled in Jupyter notebook with all you code and model performance analyses. Note that the code should be clear and readable. Make use of comments and docstrings and use functions where applicable. Source for good practices are listed above. Readability of the code will be part of the grading!
* A detailed LLM report (following the provided template), in case you have relied on language models such as ChatGPT, and Copilot.

**Grading**

You will be graded according to the following criteria:

* Working data augmentation methods (specific to your choice) with several attempts at improving model performance (show these attempt and their results). Creativity and trial end error approaches are encouraged even if they not necessarily lead to better results but remember that the goal is always to improve model performance so keep all trials sensible and well argued. (50% of the total number of points)
* Quality of the model evaluation and performance visualizations (30%)
* Quality of code (clarity, extent, and clarity of comments) (20%)