BMI / CS 771 Fall 2025:

Homework Assignment 1

September 2025

[cliu768@wisc.edu](mailto:cliu768@wisc.edu)

**Acknowledgment of AI Assistance**

This assignment was completed with the assistance of OpenAI’s ChatGPT (GPT-5). The tool was used throughout the process to brainstorm ideas, clarify relevant concepts, explain code logic, and check grammar. All final decisions regarding content, analysis, and conclusions were made by the author.

1. **Discuss the results of your transformed images**

A collage of different images of animals

Description automatically generated

This image shows the results of applying a series of random "transformations" to the original pictures. In simple terms, it's like taking one photo and creating many slightly different versions of it. By learning from all these variations, the AI gets better at recognizing an object, no matter if it's seen up close, from a different angle, or in different lighting.

1. **Explain your implementation for image rotation**

Pick a random rotation angle and rotate the image around its center (black borders appear).

Compute cosine/sine of the angle to find the largest axis-aligned rectangle that fits entirely inside the rotated image.

Handle the 45° case separately (gives a square).

Center-crop that rectangle, removing all black border pixels.

1. **Address questions in Sec. 3.3**

**3.1 What is the beneﬁt of PyTorch dataloader? Please compare the dataloader to a solution that simply loops over a ﬁle list and batches the images.**

Speed: DataLoader can load data in parallel using multiple workers, while a simple loop loads one by one.

Shuffling: It can shuffle the data for you, which helps training. A simple loop would need extra code.

Batching: It automatically groups data into batches, while a loop requires you to write this logic yourself.

In short, DataLoader saves time, avoids extra coding, and makes training smoother.

* 1. **Note that this assignment uses the same training conditions as in Tutorial 1. As part of your report, please compare the accuracy and loss trend you obtain here with those from the tutorial. Can you provide possible explanations for their diﬀerences?**

In this training run, the loss curve is noisier and does not decrease as smoothly as in Tutorial 1. For example, some batches even show a small increase after going down. This is because we used more complex image preprocessing (rotation, cropping, color changes). These transforms create more variety in the training data, which makes the task harder for the model in the short term.

In contrast, Tutorial 1 only used simple resizing. The data was more consistent, so the loss dropped more steadily.

However, both pipelines end up with similar accuracy (73%). This suggests that the stronger augmentations did not boost accuracy in this small setup, but they can help prevent overfitting and may improve generalization if we had a larger or more difficult dataset.

* 1. **Does the order of image transforms make a diﬀerence in the data augmentation pipeline? E.g., can you arbitrarily switch their order?**

Yes — the order of image transforms **does matter**.

For example, in the training pipeline:

1. Scale(72) makes the image larger first.
2. RandomHorizontalFlip() flips the whole image left–right.
3. RandomSizedCrop(64, …) then randomly cuts out a region.
4. ToTensor() finally converts it into a tensor for the model.

If we switched the order (e.g., crop first, then scale, or tensor first, then crop), the output could be very different:

* Cropping before scaling changes which part of the image is kept.
* Flipping after cropping might flip only a small patch, instead of the full image.
* Converting to tensor too early would break later transforms that expect an image.

So transforms cannot be arbitrarily reordered.

**Bonus: Max Area Rectangular after Rotation Proof**

1. **Establish a Coordinate System**: We place the center of both the original rectangle (w x h) and the cropped rectangle (cw x ch) at the origin (0, 0).
2. **Analyze a Vertex:** Due to symmetry, we only need to analyze one vertex of the cropped rectangle. Let's choose the vertex in the first quadrant, P′=(cw/2,ch/2).
3. **Apply Inverse Rotation:** We apply an inverse rotation of angle −θ to point P′ to find its corresponding location, P=(x,y), in the original, un-rotated space.

x=2cw​cosθ+2ch​sinθ

y=−2cw​sinθ+2ch​cosθ

1. **Define the Constraints:** For the point P=(x,y) to be inside the original w x h rectangle, its coordinates must satisfy:

∣x∣≤w/2 and ∣y∣≤h/2

Substituting the expressions for x and y and simplifying, we get the two key linear inequalities that constrain cw and ch (assuming θ is between 0 and 90 degrees for simplicity):

cw⋅cosθ+ch⋅sinθ≤w

cw⋅sinθ+ch⋅cosθ≤h

The goal is to **maximize the area function A=cw⋅ch.** The vertices of our feasible region are the points where the boundary lines intersect.The most important vertex is the intersection of the two main constraint lines.Finding this vertex is therefore equivalent to solving the system of linear equations where the inequalities become equalities:

**cw= (w⋅cosθ−h⋅sinθ)/cos(2θ)**

**ch=(h⋅cosθ−w⋅sinθ)​/cos(2θ)**