Understanding Regularization in CNNs Using the EuroSAT Dataset: A Practical Tutorial

(GitHub Repo: https://github.com/vgowtham009/eurosat-regularization-tutorial.git)

1. Introduction

Deep learning models—especially convolutional neural networks (CNNs)—are remarkably good at recognising visual patterns. They can distinguish cats from dogs, diagnose diseases from X-rays, and even classify satellite images taken hundreds of kilometres above Earth. However, this power comes with a well-known weakness: CNNs love to memorise. When the amount of training data is limited or the patterns are subtle, they often learn the training set "too well," performing brilliantly on familiar examples but struggling on new ones. This phenomenon, overfitting, is one of the central challenges in machine learning.

Regularization techniques exist to counter exactly this problem. They nudge the model away from memorisation and toward learning stable, generalisable representations. Some do this by penalising large weights, others by removing neurons during training, and some by artificially "expanding" the dataset. Each method tackles overfitting from a different angle, and understanding these angles is key to building robust models.

In this tutorial, we investigate these techniques using the EuroSAT RGB dataset—a real-world collection of satellite images covering ten land-use categories such as forests, residential areas, and agricultural fields. Although EuroSAT contains over 27,000 images, we deliberately constrain ourselves to only 30% of the dataset for training. This intentional limitation creates a realistic, data-scarce scenario where overfitting becomes easy to observe and regularization becomes essential. In other words, we create the perfect environment to study the behaviour of regularization.

This leads us to our question: How do different forms of regularization—weight decay, dropout, data augmentation, batch normalization, and early stopping—shape the learning behaviour and generalisation performance of CNNs on EuroSAT under limited data conditions?

To explore this, we keep the CNN architecture fixed while systematically applying different regularization methods, both individually and in combination. We then analyse how these choices affect model training curves, validation behaviour, and final test accuracy.

By the end of this tutorial, the goal is not only to identify which technique performs best, but to uncover why certain strategies are more effective for satellite imagery and when each should be used in practice. Ultimately, this exercise aims to deepen intuition around regularization—one of the most important, yet often misunderstood, aspects of modern deep learning.

2. Dataset & Experimental Setup

For this tutorial, we use the EuroSAT RGB dataset, a collection of satellite images representing ten different land-use categories such as forests, residential areas, lakes, and agricultural fields. Each image is small (64×64×3), but visually rich, making the dataset a surprisingly good testing ground for understanding when CNNs overfit and how regularization helps.

To make the problem intentionally challenging—and to expose overfitting more clearly—we train on only 30% of the dataset, keep 10% for validation, and reserve the remaining 60% for testing. This

limited-data setup mirrors real scenarios where labelled satellite imagery is scarce, and it gives our regularization experiments something meaningful to fix.

Every experiment uses the same CNN backbone: three convolutional blocks with ReLU activations and max-pooling, followed by a dense layer and a softmax classifier. What changes between experiments is only the type of regularization applied. This ensures that any change in performance is due to the technique itself rather than differences in model size or optimisation strategy.

All models are trained with the Adam optimizer, a batch size of 64, and up to 25 epochs. For experiments involving early stopping, training ends automatically once the model stops improving on the validation set.

By holding everything else constant, we create a clean, controlled environment to observe how different regularization strategies influence learning—and more importantly, why certain techniques perform better on satellite imagery than others.

3. Regularization Techniques

When a CNN is given limited data—as in our 30% EuroSAT setup—it tends to latch onto every tiny detail it sees, even the ones that don't matter. Regularization techniques act like gentle reminders to the model: "Don't get too confident. Learn the pattern, not the picture." Each method approaches this problem from a different angle.

L2 Weight Decay — Controlling the "size" of the model's beliefs L2 regularization adds a small penalty to large weights, discouraging the network from relying too heavily on any single feature. Instead of memorising exact pixel arrangements, the model is nudged to learn smoother, more general patterns. Mathematically it's simple, but its effect—reducing overfitting without changing architecture—is surprisingly powerful.

Dropout & SpatialDropout2D — Teaching the model to rely on everyone Dropout randomly "switches off" neurons during training. This prevents the network from depending on a specific pathway and forces it to distribute what it learns. SpatialDropout2D takes this a step further by dropping whole feature maps, which is especially helpful in CNNs where entire channels can become overly specialised.

Data Augmentation — Growing the dataset without collecting anything new Data augmentation creates new training examples by randomly flipping, rotating, zooming, or adjusting the contrast of images. For satellite imagery, this is incredibly effective—landscapes look different depending on angle, lighting, and terrain. Augmentation teaches the model that forests are forests whether viewed from slightly left, slightly right, bright, or dim. This is often the strongest regularizer because it expands the model's experience without expanding the real dataset.

Batch Normalization — Stabilising learning as the model grows Batch normalization normalises activations inside the network, keeping them from drifting into ranges that make learning unstable. This stabilizing effect has a side benefit: it acts as an implicit regularizer, making the model less sensitive to small changes in the training data and smoothing out the optimization landscape.

Early Stopping — Knowing when to quit Sometimes the best regularization is simply not letting the model get carried away. Early stopping watches the validation loss and ends training when the model begins to overfit. It's a practical, training-time regularizer that saves both performance and computation.

Together, these techniques offer a full toolbox for controlling overfitting—from weight penalties and neuron dropout to data expansion and training-time safeguards. The goal of our experiments is to see how each one behaves on the EuroSAT dataset and what they reveal about learning patterns in satellite imagery.

4. Experiments

To understand how different regularization strategies influence generalisation on EuroSAT, we trained the same CNN architecture under six controlled configurations. Each experiment changes only the regularization method; everything else—data splits, optimizer, batch size, number of epochs—remains identical. This setup lets us isolate the effect of each technique without confounding variables.

Experiment 1 — Baseline (No Regularization) This model serves as the reference point. It has no weight decay, no dropout, no data augmentation, and no training-time tricks. With limited training data, the baseline is expected to overfit heavily. Purpose: Show what the model does when nothing holds it back.

Experiment 2 — L2 Weight Decay Only Here we apply a small L2 penalty to all convolutional and dense layers. This discourages large weights and subtly nudges the model toward smoother, less overfitted solutions. Purpose: Test whether weight-space regularization alone can reduce overfitting in satellite imagery.

Experiment 3 — Dropout + SpatialDropout2D This configuration focuses on activation-level regularization. Dropout randomly removes neurons in the dense layer, while SpatialDropout2D removes entire feature maps within convolutional layers. Purpose: See how preventing co-adaptation affects the CNN's ability to generalise across EuroSAT classes.

Experiment 4 — Data Augmentation Only We apply randomized flips, rotations, zoom, and contrast changes to each batch. No dropout, no L2. Purpose: Evaluate how expanding the visual diversity of the dataset improves robustness—especially important for satellite images that naturally vary in orientation and lighting.

Experiment 5 — Batch Normalization + Early Stopping Batch normalization stabilizes internal activations, while early stopping halts training once validation loss stops improving. Purpose: Explore implicit regularization (BN) and training-time regularization (ES) as a combined approach.

Experiment 6 — Strong Combination Model This final model combines several techniques: L2, dropout, SpatialDropout2D, data augmentation, batch normalization, and early stopping. In theory, combining multiple methods should produce the most robust model—though in practice, too much regularization can limit capacity. Purpose: Investigate how regularizers interact when stacked together, and whether "more" truly equals "better."

5. Results & Interpretation

The experiments reveal clear differences between models. The baseline model nearly memorises the training set, reaching close to 99% training accuracy, while validation accuracy plateaus around 0.8, demonstrating classic overfitting. The strong-combination model increases more slowly and caps lower, indicating underfitting due to excessive regularization.

The loss curves tell a similar story. Baseline validation loss decreases briefly but then rises steadily, another sign of overfitting. The strong-combination model maintains higher loss but with reduced divergence, reflecting reduced model capacity.

When comparing all six models, data augmentation and BatchNorm+EarlyStopping show the most consistent and highest validation accuracy, outperforming L2, dropout, and the strong-combination model. This suggests that data-space and training-time regularization are especially effective for EuroSAT's natural variability in orientation and illumination.

Test accuracy for each experiment is as follows: Baseline: 0.7942 L2 Only: 0.8032 Dropout + Spatial: 0.8148 Data Augmentation: 0.8609 BatchNorm + EarlyStopping: 0.8642 Strong Combo: 0.8178

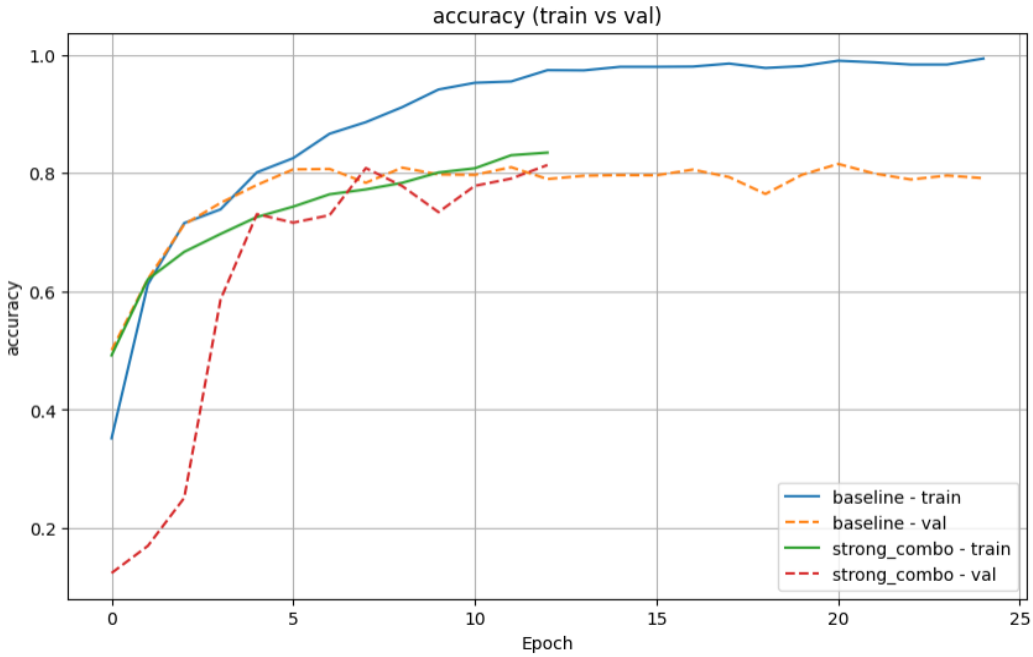

Figure 1. Training and validation accuracy for the baseline and strong-combination CNNs.

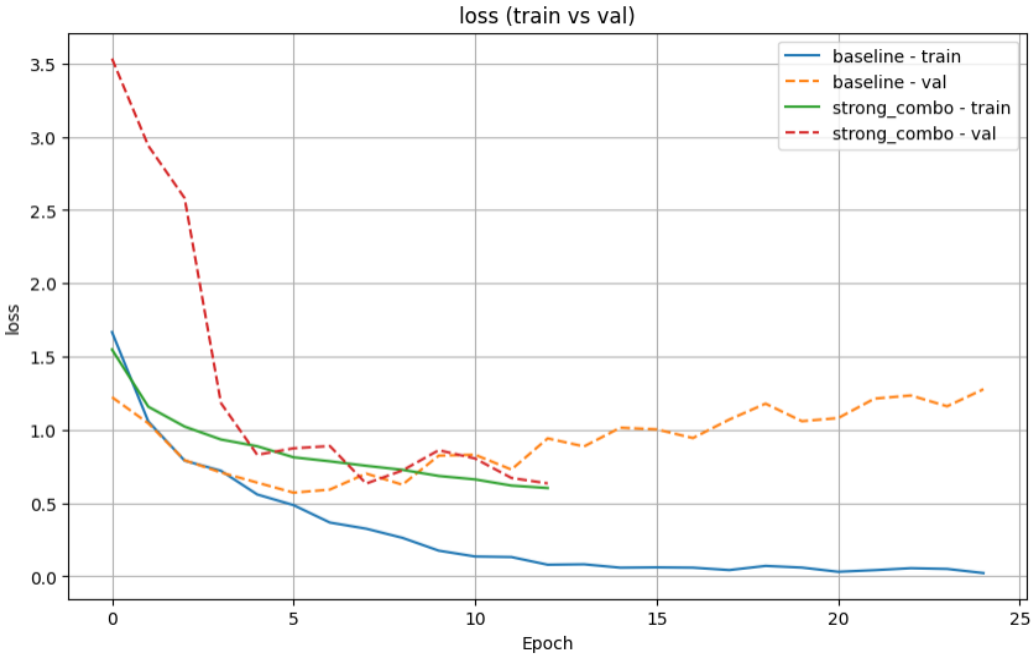

Figure 2. Training and validation loss for the baseline and strong-combination models.

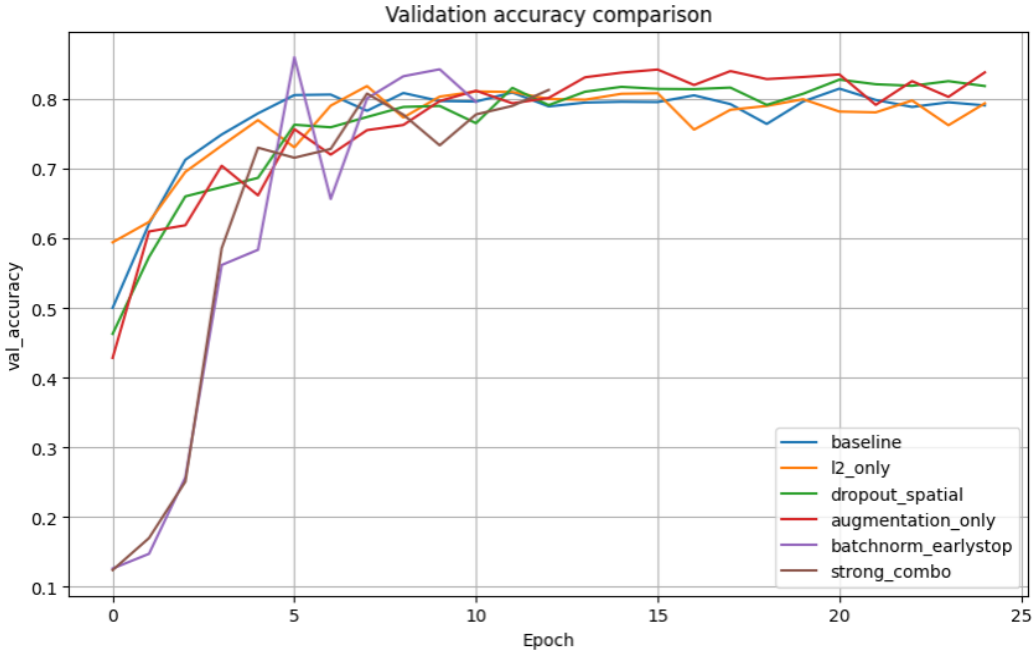

Figure 3. Validation accuracy across all six regularization methods.

Experiment	Test Accuracy
Baseline	0.7942

L2 Only	0.8032
Dropout + Spatial	0.8148
Data Augmentation	0.8609
BatchNorm + EarlyStopping	0.8642
Strong Combo	0.8178

6. Discussion

The experiments reveal that regularization is not a one-size-fits-all solution; each method contributes differently depending on the characteristics of the dataset and model. EuroSAT, despite its relatively small image size, contains complex spatial patterns that vary in orientation, texture, and illumination. As a result, the model benefits most from data-space and training-time regularization, which expose it to a wider variety of examples and stabilise the optimisation process.

In contrast, weight-based and activation-based regularization—such as L2 and dropout—offer modest improvements but cannot fully compensate for the limited diversity in the training split. These methods prevent extreme weight values and co-adaptation, but they do not expand the set of visual patterns the model encounters.

An interesting outcome is that the strongest combined model did not perform the best. This highlights an important practical lesson: applying too many regularizers can restrict the model's capacity to learn meaningful features, especially in datasets with subtle class boundaries like EuroSAT. Regularization needs to be balanced, not maximised.

Ultimately, these results emphasise that understanding the nature of the data and the goal of the model is essential when selecting regularization strategies. Data augmentation and batch normalization succeed here because they address the kinds of variation that naturally occur in satellite imagery.

7. Conclusion

This tutorial explored how different regularization strategies affect CNN performance on the EuroSAT RGB dataset under intentionally limited training conditions. Starting from a highly overfitted baseline, we demonstrated how weight decay, dropout, data augmentation, batch normalization, and early stopping each influence generalisation in distinct ways.

The strongest individual improvements came from data augmentation and batch normalization paired with early stopping, which together produced stable learning dynamics and high test accuracy. Conversely, overly aggressive combinations of regularizers reduced model capacity and led to underfitting, reinforcing the idea that regularization must be applied thoughtfully.

Overall, this study shows that effective regularization is not just about preventing overfitting—it is about guiding the model toward representations that reflect how the real world varies. For satellite imagery, encouraging invariance to rotation, illumination, and texture proves far more impactful than penalising individual weights.

References

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. JMLR.
Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training. ICML.
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep CNNs. NeurIPS.
Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). EuroSAT: A Novel Dataset and Deep Learning Benchmark. IEEE JSTARS.