Report: Star and Streak Detection in Astronomical Images

By: Ashwathy Nair (nairashwathy99@gmail.com)

Observations and Conclusions (Question 1)

1. Introduction

This project aims to detect and classify stars and streaks in grayscale astronomical images captured by a telescope. The goal is to build a convolutional neural network (CNN) model that classifies image patches into three classes: star, streak, or noise/empty. A manual labelling step was performed to prepare training data, followed by model training and evaluation. Additionally, centroid detection of stars and streaks was attempted on predicted patches.

2. Data Preparation

Images were loaded from a collection of 16-bit grayscale TIFF files. Each image was normalized by scaling pixel intensities to the [0,1] range. Random patches of size 128x128 pixels were extracted from these images to create the dataset for classification. A total of approximately 175 patches were generated, with 100 patches manually labelled for training and the rest reserved for testing and prediction.

3. Manual Labelling

Manual labelling was done using an interactive widget to classify patches into star, streak, or noise/empty categories. During this step, only star and noise patches were clearly identified. No streaks were labelled, possibly due to their faint appearance or rarity in the dataset. This imbalance in class labels impacted subsequent model training and prediction.

4. Model Development

A CNN model was designed with three convolutional layers followed by max-pooling, flattening, and fully connected dense layers. The output layer used softmax activation to classify patches into three classes. The model was trained on manually labelled patches with sparse categorical cross-entropy loss and the Adam optimizer. Due to the limited and imbalanced dataset, the model likely overfitted, as indicated by near-perfect training and validation accuracy.

5. Results

The model achieved an accuracy of 1.0 during training, but this metric did not reflect true model performance. On testing, the model predicted only the "noise" class and failed to detect any stars or streaks, despite the presence of stars in the manually labelled data. This indicates that the model **overfitted** to the dominant class (noise), likely due to a small and imbalanced dataset (Kim & Brunner, 2017).

6. Centroid Detection

Centroid detection was implemented by thresholding patches labelled as stars or streaks and computing the centre of mass of the bright regions. However, since all predicted labels were noise, no star or streak centroids could be extracted from the predictions. Manual labelling data showed some star centroids, but streak centroids were absent due to the lack of streak labels.

7. Challenges & Assumptions

- **Manual Labelling Quality:** Difficulty in visually identifying faint streaks likely led to zero streak labels, causing class imbalance.
- **Dataset Size:** The limited number of manually labelled patches constrained the model's ability to generalise.
- **Image Quality:** Variability and faintness of streaks made them hard to detect manually and by the model.
- **Assumptions:** It was assumed that manual labelling was accurate and that random patch extraction would capture representative samples of stars and streaks.

8. Conclusion

This project highlights the challenges of detecting faint streaks in astronomical images using limited labelled data. The CNN model overfitted the training data, leading to poor generalisation and an inability to detect streaks during prediction. Future work should focus on improving manual labelling with better visualisation techniques, increasing dataset size, and employing data augmentation or transfer learning to enhance model robustness.

Question 2:

a) **How can the model achieve high accuracy when the available training images are insufficient? What strategies can be used to overcome this limitation?**

In this project, the dataset was quite small, and most of the patches contained only stars or noise and no streaks were manually labelled, which limited what the model could learn. When training data is limited, it's common for the model to overfit, especially in classification tasks like this.

To improve performance despite limited data, a few strategies can help:

- <u>Data augmentation</u> (e.g., rotation, flipping, adding noise) creates more diverse training examples.
- <u>Transfer learning</u> using pre-trained CNNs can help the model learn useful features from larger datasets and adapt them to astronomy images.

- Semi-supervised learning might also help by using confident model predictions on unlabelled data to expand the dataset.

Overall, working with such a small and skewed dataset made it hard for the model to generalise, and that's something that needs to be addressed in future steps.

b) How can model overfitting and underfitting be addressed? What factors contribute to these issues?

In this case, the model showed 100% accuracy on training/validation data, but all test predictions came out as “noise.” This is a classic sign of overfitting where the model memorised the training data rather than learning general patterns.

To reduce overfitting:

- We can apply dropout layers or regularisation during training.
- Simplifying the model architecture (fewer layers or filters) can help when working with limited data.
- Early stopping can prevent training for too long on noisy or redundant data.

Underfitting would happen if the model performs poorly on both training and test sets, which wasn’t the case here.

Contributing factors in our case include:

- Very few training samples.
- Missing classes (no streaks in labels).
- High similarity between noise and faint features.
- The CNN being too sensitive to brightness patterns, misclassifying stars or streaks as background.

c) Can the developed model accurately detect streaks and stars in real-sky astronomical images, given varying signal-to-noise conditions (e.g., faint object detection)? How does this differ from simulated images?

Given the current state of the model, it’s unlikely to perform well on real-sky telescope images. The model struggled even with synthetic images, it learned to predict everything as noise, likely due to missing examples of streaks and very few star patches.

Real-sky images are much harder to process because they contain:

- Varying signal-to-noise ratios.
- Faint streaks that are hard to distinguish from background stars.
- Atmospheric distortion and other types of noise.

Unlike simulated images, real ones aren't clean (features blend in more), and the model needs to be robust to contrast and brightness changes

To improve, the model would need:

- Better preprocessing (e.g., denoising, contrast stretching).
- A dataset with well-labelled examples of both bright and faint features.
- Possibly combining CNNs with classical image processing methods (like line detectors) for streaks.

d) What is the likelihood of false detections, such as detecting multiple blobs along a streak instead of a continuous line? Will the model correctly report false positives and generate an appropriate error matrix?

In the current version, false detections are very likely, especially since the model labelled all test patches as "noise." Even if it had predicted stars or streaks, there's a good chance it would have picked up small blobs or bright regions and missed the continuity of actual streaks.

This happens because:

- Patches are small and don't capture full streaks.
- The model doesn't understand spatial continuity, which means it looks at brightness patterns locally.
- No streaks were included in the training data, so the model never learned what to look for.

As for false positives/negatives and the confusion matrix, the model will generate an error matrix only if it predicts multiple classes. Since all predictions were "noise," the matrix was essentially one-sided, showing that the model didn't detect any stars or streaks. So, right now, it can't report false positives accurately.

To reduce false detections, post-processing steps like morphological analysis or Hough transforms could help detect line-like features, especially in full images rather than small patches.

REFERENCES

Kim, E. J., & Brunner, R. J. (2017). Star–galaxy classification using deep convolutional neural networks. *Monthly Notices of the Royal Astronomical Society*, *464*(4), 4463–4475. https://doi.org/10.1093/MNRAS/STW2672