Tracking Issues for Research Projects

1. Purpose of Issues

The **Issues** section (or tracker) is meant to **log anything that needs attention**.

For research projects, this includes:

- Experiments that failed – e.g., *"model overfitting on dataset X"*
- Ideas for improvement – e.g., *"try different learning rate"*
- Feature requests – e.g., *"add early stopping in training script"*
- Data problems – e.g., *"missing values in column Y"*

> Think of **issues as your project memory** — a place to store observations, tasks, and problems so you don't forget.

---

2. How to Manage Issues

You can use **GitHub Issues**, **Notion**, **Trello**, or even a simple **Markdown file** in your repository.

---

a) Use Labels

Labels help categorize issues for easy filtering:

Label	Purpose
`bug`	Something in code or script isn't working as expected
`experiment`	A note about an experiment that needs running, failed, or needs tweaking
`feature`	New function, script, or model improvement to add
`dataset`	Dataset problem, missing data, preprocessing step needed

---

b) Record Observations Clearly

For each issue, include:

1. **Title:** Short summary
   Example:
2. `Model overfitting on dataset X`

3. `Try different learning rate for CNN model`

4. **Description:** Detailed explanation

5. What you tried
6. What went wrong or what you observed
7. Possible next steps

Example Markdown:

```
### Title: Model overfitting on dataset X
**Type:** experiment
**Observation:** Training accuracy reaches 99% but validation accuracy is only
70%.
**Next steps:**
- Try reducing learning rate
- Add dropout
- Collect more training data
```

---

c) Benefits

- Keeps **all problems and ideas in one place**
- Makes it easy to **resume work later**
- Helps **collaborators understand issues** quickly
- Makes your project **organized and professional**

> **Tip:** Think of issues like a "lab notebook for coding experiments." Open a new issue every time something fails or you have a new idea, and close it when resolved.