A brief on the approach, which you used to solve the problem.

This was my first time participating in any deep learning hackathon. Give that there was ample time for the hackathon I followed Jeremy Howards FastAI course, and implemented all the learning here. For my solution I used

- Base DenseNet201 architecture pre-trained on ImageNet dataset for transfer learning
- Added some more layers to the base architecture.
- Used Differential and Cyclic learning rates (Implemented by default in FastAI)
- Data Augmentation using different types of transformation (Zoom, Crop, Rotate, Resize, Squish, Perspective Wrap, Symmetric Wrap etc.)
- Used Test Time Augmentation to improve the predictions on Test Set
- Optimized TTA for different scale with fixed set of augmentations (for replication) which also gave the best results
- Used Snapshot ensembling.
- Trained on different size of images.

Which data-preprocessing ideas really worked? How did you discover them?

There wasn't much of data-preprocessing needed for this problem. The only one I used is:

- Resizing the image to square boundaries. Recommended by multiple medium blogs and fastai course.

Mention the pre-trained models with links used for building the model

DenseNet201 pre-trained on ImageNet Dataset
https://pytorch.org/docs/stable/torchvision/models.html

How does your final model look like? How did you reach it?

Ensemble of multiple snapshots of DenseNet201 with additional layers. Github and blogs.

What are the key takeaways from the challenge, if any?

- You don't need much data for deep learning when you can use transfer learning and Data Augmentation Tricks
- Using Test Time Augmentation to boost model performance on Test Set

According to you, what are the 5 things a participant must focus on while solving such problems?

1. Read about SOTA approaches for the problems with similar domains.
2. Transfer Learning is a way to go.
3. Tricks like Data Augmentation and TTA must always be tried. They help the model generalize better.
4. Try multiple Architecture, pre-trained models, customized pre-trained models.
5. Keep Learning and Reading.