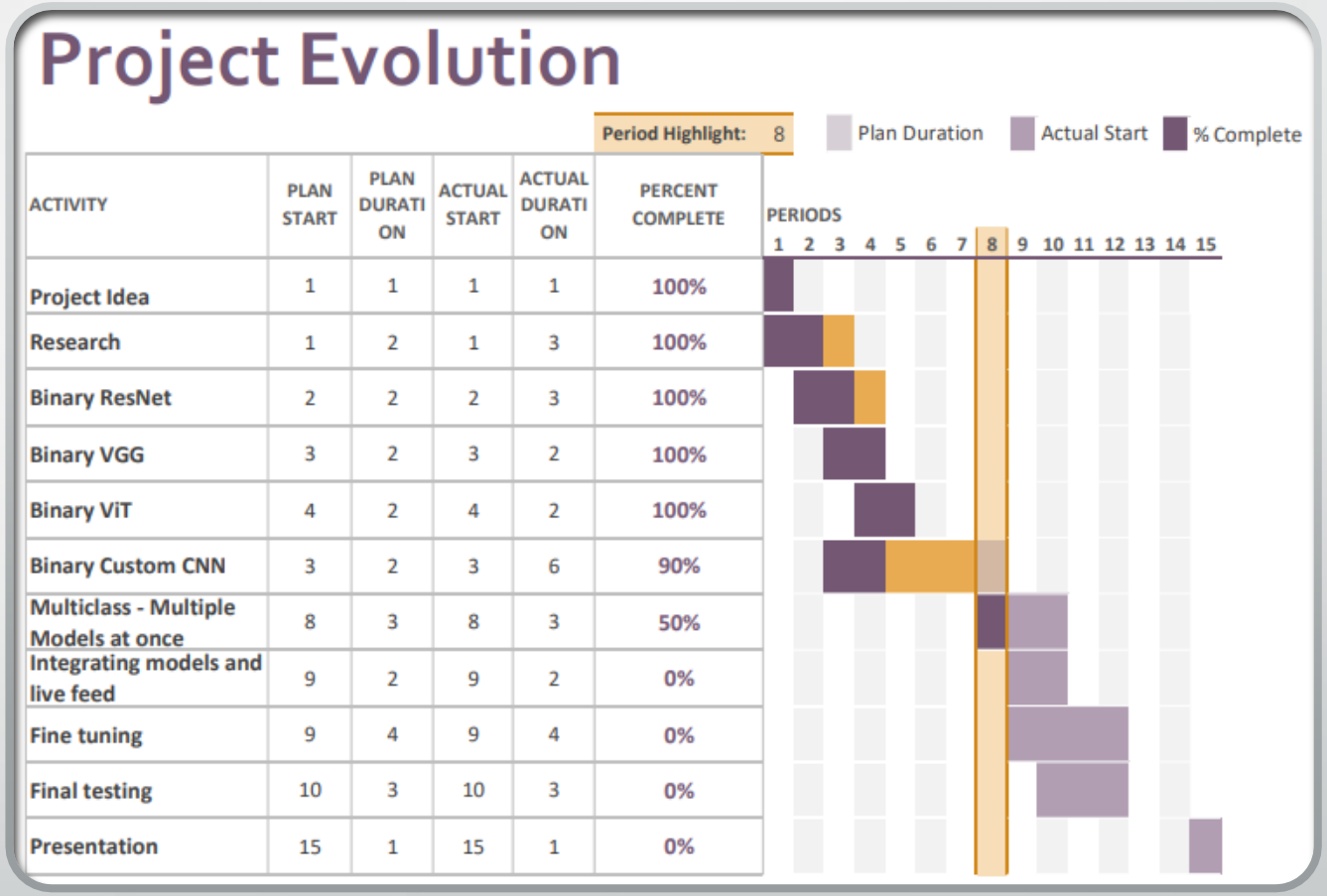
Classification of Dry Eyes Through Meibography Demo 1

Arnar Evgení Gunnarsson, Magnús Benedikt Magnússon,
Ragnheiður Gná Gústafsdóttir

T-869-COMP - Computer Vision

Project evolution

- Initial concept still on:
  - Develop an image recognition system to determine meibomian gland health.
  - Progress so far only applicable to our initial minimum goal.
- Progress:
  - Datasets modified to accommodate both binary and spectrum evaluations.
  - Codebase working with a couple of models, including training, validation and testing.
  - Models trained, though only on the binary datasets.
  - Testing of models not started.



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Project Idea	1	1	1	1	100%
Research	1	2	1	3	100%
Binary ResNet	2	2	2	3	100%
Binary VGG	3	2	3	2	100%
Binary ViT	4	2	4	2	100%
Binary Custom CNN	3	2	3	6	90%
Multiclass - Multiple Models at once	8	3	8	3	50%
Integrating models and live feed	9	2	9	2	0%
Fine tuning	9	4	9	4	0%
Final testing	10	3	10	3	0%
Presentation	15	1	15	1	0%

Background material

- **Papers:** A Deep Learning Approach for Meibomian Gland Atrophy Evaluation in Meibography Images (Jiayun Wang et al.), Automated quantification of meibomian gland dropout in infrared meibography using deep learning (Ripon Kumar Saha et al.), Multi categorical of common eye disease detect using convolutional neural network: a transfer learning approach (Abu Kowshir Bitto, Imran Mahmud).
- **Existing systems:** No commercial systems available.
- **Tools:** Intel RealSense D455 and phone cameras.
- **Software:** PyTorch, fun experiment on TensorFlow.
- **Datasets:** In-house RU dataset, MGD-1k, Meibographies Pult.

Methods Tested

- Various methods considered for the project.
  - Frameworks:
    - YOLO – Object Detection
    - Torchvision – Part of PyTorch ecosystem.
    - Keras and TensorFlow – Ease of use when generating custom model.
  - Datasets:
    - Train on specific set or mixed sets:
      - In-house RU dataset: Not balanced, more healthy than unhealthy.
      - MGD-1k: Well documented and more balanced dataset.
      - Meibographies Pult: Sourced in-house.

Methods Selected

- Pivoted from YOLOv5 to Torchvision
  - Object detection vs image classifications.
  - Meticulous adjustments needed vs. flexibility
  - Model variety within Torchvision
- Custom CNN with Keras and TensorFlow
  - Supposed ease of use when generating custom models.
  - Just out of curiosity.
- Mixed datasets
  - To enhance the robustness of trained models.

Performance Measures

```
Model:   VGG16
Correctly classified: 85    |Incorrectly classified: 15 |Accuracy Score: 85.0
Class:  grade_0
Precision 0.8    Recall 0.9824561403508771
Class:  grade_3
Precision 0.9666666666666667     Recall 0.6744186046511628
```

```
Model:   ResNet50
Correctly classified: 84    |Incorrectly classified: 16 |Accuracy Score: 84.0
Class:  grade_0
Precision 0.7971014492753623     Recall 0.9649122807017544
Class:  grade_3
Precision 0.9354838709677419     Recall 0.6744186046511628
```

```
Model:   ViTB32
Correctly classified: 80    |Incorrectly classified: 20 |Accuracy Score: 80.0
Class:  grade_0
Precision 0.8032786885245902     Recall 0.8596491228070176
Class:  grade_3
Precision 0.7948717948717948     Recall 0.7209302325581395
```

Live Demonstration

Arnar's awesome live demo!

Work Involved

- Division of labour - who is doing what?
  - Arnar: Code Master & Model Trainer
  - Magnús: Coding, ResNet and Documentation
  - Ragnheiður: Coding, Custom CNN and Documentation
- What has been most time-consuming?
  - Package management.
  - Constructing codebase.
  - Training models.
- Particular challenges faced.
  - Size and Quality of the datasets.

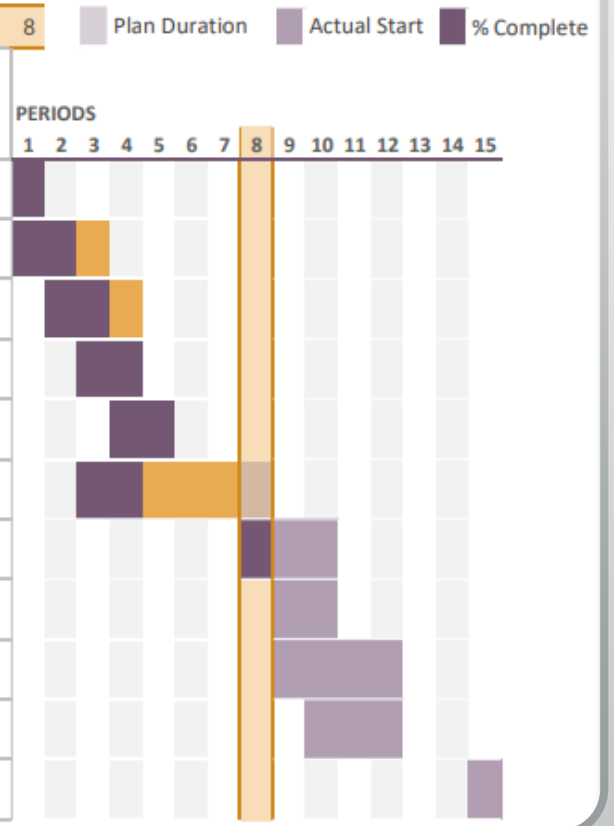
Plan Going Forward

- Our goal for Demo 2 next week is to be able to show you a live feed of binary classification and with more than random accuracy (25%).

Project Evolution

Period Highlight: 8 | Plan Duration | Actual Start | % Complete

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Project Idea	1	1	1	1	100%
Research	1	2	1	3	100%
Binary ResNet	2	2	2	3	100%
Binary VGG	3	2	3	2	100%
Binary ViT	4	2	4	2	100%
Binary Custom CNN	3	2	3	6	90%
Multiclass - Multiple Models at once	8	3	8	3	50%
Integrating models and live feed	9	2	9	2	0%
Fine tuning	9	4	9	4	0%
Final testing	10	3	10	3	0%
Presentation	15	1	15	1	0%



Questions?