Project: healthcare timeseries

Antonio Alampi - 64316 Tommaso Tragno – 64699 Cristian Tedesco – 65149 Pol Rion Solé – 65177

A. Quick rubric (✓ / ▵ / ✗)

Project	Technical soundness	Feasible on free-tier GPUs	Dataset availability & readiness	Starting-code / transfer-learning plan	Evaluation metrics
Early Alzheimer's (multivariate time-series + neuroimaging)	▵ – conceptually plausible but multimodal fusion plan is still vague	✗ – MRI preprocessing & 3-D models will exceed Colab/Kaggle limits	▵ – NACC access request, heavy cleaning	▵ – few open LSTM + MRI stacks; will need custom code	✓ – accuracy, precision, sensitivity, ROC
Early Parkinson's (speech)	✓ – CNN + LSTM on spectrograms is a proven pipeline	✓ – audio models train in < 4 GB VRAM	✓ – Parkinson's Voice Initiative & PhysioNet are open	✓ – many public repos; Wav2Vec2 fine-tuning possible	✓ – sensitivity, specificity, ROC
Facial fatigue (images / short clips)	✓ – single-frame CNN with temporal smoothing is standard	▵ – videos large, but frame-based transfer learning fits 8 GB GPU	✓ – NTHU-DDD freely downloadable	✓ – drowsiness-detection codebases available	✓ – accuracy, ROC, Grad-CAM

(✓ = ready; ▵ = needs work; ✗ = high risk)

B. Targeted suggestions

1. Early Alzheimer's detection

- **Narrow the modality scope.** Start with the tabular biomarker & cognitive-score subset (CSV) before adding MRI. This removes 90 % of the preprocessing burden and still supports LSTM time-series modelling.
- **Use tabular DL baselines.** Fast TabNet or simple GRU stacks give a quick performance reference; migrate to more complex deep-stacking only if baselines plateau.
- **Storage & VRAM checks.** A single 3-D MRI volume (~150 MB) × thousands of patients will break Colab quotas. If neuroimaging is essential, slice MRIs to 2-D and use transfer-learning on ImageNet-pre-trained EfficientNet.
- **Access timeline.** NACC approval typically takes ≥ 2 weeks; submit the request immediately if you keep this topic.

2. Early Parkinson's detection via speech

- **Leverage self-supervised audio models.** Fine-tune *Wav2Vec 2.0* or *TRILL*; this reduces training time and lifts accuracy over handcrafted-spectrogram CNNs.
- **Automate preprocessing.** Use *librosa*'s pipeline (resample → normalize → Mel-spectrogram) inside a tqdm script so data prep runs once and caches to disk.
- **Balanced splits.** Parkinson's datasets are class-imbalanced; stratified k-fold cross-validation plus class-weighting or focal loss is advisable.
- **Proof-of-concept first.** Train a 1-D CNN on raw MFCCs to set a baseline within a single Colab session (< 1 h). Iterate from there.

3. Facial fatigue detection

- **Frame-based over video-based.** Extract one frame per second; train an ImageNet-pre-trained EfficientNet-B0. Add temporal smoothing (e.g., 5-frame median) only at inference. Saves GPU RAM.
- **Subject-wise split.** Avoid data leakage: ensure all frames of a participant sit in the same fold.
- **Lightweight deployment.** If you aim for real-time, export to *ONNX* or *TensorFlow Lite* and profile on CPU.
- **Augmentation.** Eye-region crops, random brightness, and horizontal flips help generalize to illumination changes.

General advice

- **Choose one project now.** The Parkinson's-speech track is the least risky given data availability and GPU limits.
- **Document preprocessing scripts.** Push a minimal `prepare_data.py` to the repo this week so later milestones focus on modelling.
- **Track compute time.** Keep a log of Colab GPU hours; aim for < 5 h to reach a first benchmark so you have slack for hyper-parameter tuning.
- **Metrics first.** Implement evaluation (ROC-AUC, confusion matrix) before complex modelling; this safety-checks the pipeline end-to-end.

Transfer learning advice:

Transfer-learning guidelines tailored to your three candidate topics

Project	Best starting weights	Fine-tuning strategy	Practical tips for free-tier GPUs
Early Parkinson's (speech)	• *Wav2Vec 2.0* base (Facebook) or *Whisper-tiny* (OpenAI)• Lightweight fallback: *YAMNet* (MobileNet-based) on log-Mel features	1. Replace the pre-training CTC/head with a two-layer MLP for binary PD vs. control.2. **Phase 1**: freeze the encoder, train only the new head (3–5 epochs, LR≈3e-4).3. **Phase 2**: unfreeze the top 20 %	• Trim clips to 5-10 s to fit batch size 8 on 8 GB VRAM.• Gradient accumulation ×2 emulates batch 16 without extra memory.• SpecAugment still helps even with SSL models.

Project	Best starting weights	Fine-tuning strategy	Practical tips for free-tier GPUs
		of transformer blocks, use a discriminative LR schedule (head > unfrozen > frozen).	
Facial fatigue (images)	• *EfficientNet-B0* or *MobileNetV3-Large* (ImageNet).• If eye cues are dominant, consider *Vision-Ocular ViT* weights from FaceQnet v3.	1. **Phase 1**: freeze backbone, train a 256-unit ReLU + Softmax head.2. **Phase 2**: unfreeze last two MBConv/Transformer blocks; LR≈1e-4 with cosine decay.3. Optional: plug a lightweight temporal module (1-D conv over logits) trained from scratch.	• Pre-crop faces at 224×224 to keep GPU memory under 2 GB.• Use mixed-precision (`torch.cuda.amp`) and gradient checkpointing to halve VRAM.• Early stopping with patience = 5 avoids over-fine-tuning.
Early Alzheimer's (tabular + MRI slices)	**Tabular branch**: no large-scale public pre-train, but *TabNet* or *SAINT* checkpoints on Kaggle categorical datasets transfer reasonably.**Imaging branch**: *Med3D* or *MedicalNet* 3-D ResNet18 weights trained on 23 medical datasets; for 2-D slices, ImageNet ResNet50.	**Two-stage**: (i) Train each modality separately with frozen pre-training layers → obtain embeddings; (ii) Concatenate embeddings and train a shallow fusion MLP.Freeze MRI backbone entirely unless you have ≥24 GB; fine-tune only the fusion head and the last FC layer of the tabular net.	• Convert 3-D MRI to three orthogonal 2-D views (axial, coronal, sagittal) to reuse 2-D pretrained weights.• Use `torchio.transforms.CropOrPad` to standardize volumes offline—do it once on CPU.• Batch size 4, mixed precision, and slice-level training fit on Colab Tesla T4 (16 GB).

Cross-cutting recommendations

1. **Layer-wise learning-rate decay**
   Apply an exponential decay (e.g., $\gamma = 0.95$) from the new head downwards; avoids catastrophic forgetting while letting high-level features adapt.
2. **Selective regularization**
   Use weight-decay = 0 for pretrained biases + batch-norm gains; keep standard decay (1e-2) for newly initialized weights.
3. **Checkpoint-averaging over last K epochs**
   Stabilizes fine-tuned models without extra compute or memory.
4. **Re-initialization rule of thumb**
   If a pretrained convolutional layer sees input statistics far from ImageNet/audio waveform (e.g., biomarker CSVs), re-initialize or keep it frozen—otherwise gradients explode.

5. **Reproducibility**
   Pin seeds, log Git commit + hyper-parameters, and export inference-ready ONNX/TFLite models; this matters for the poster session.

Following these practices, each project remains trainable within free-tier GPU limits while exploiting mature, high-quality representations instead of spending course time on low-level feature learning.