

COVID-19 DETECTION MILESTONE 2



By: Drew Quashie, Loleyi Oluwatomisin, Amanda Ogbonna, and Richard Alonso Garcia

MILESTONE 2 GOALS

From Milestone 1's baseline to a trained, evaluated, servable system:

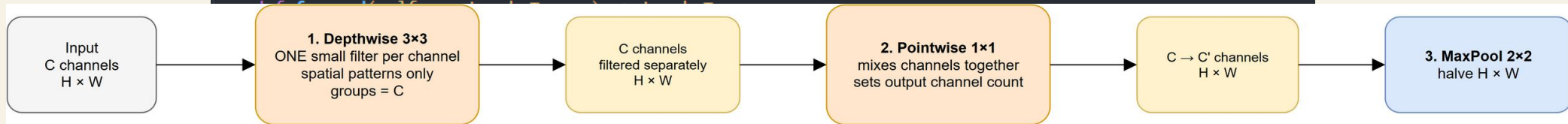
- Train & fine-tune CNN models + transfer-learning architectures (85%)
 - Amanda and Drew
- Initial Model Evaluation & Performance Reports (85%)
 - Amanda, Loleyi and Drew
- FastAPI backend endpoints (90%)
 - Drew and Loleyi
- React prediction/risk UI (70%)
 - Richard

TOOLS USED

Domain	Tool / Technique
Frontend Framework	React, Web Audio API
Backend & API	Python, FastAPI
Machine Learning	PyTorch, Librosa (Feature Extraction)
Model Architectures	Convolutional Neural Networks (CNNs), ResNet (Transfer Learning)
Dataset	CoughVid Dataset v3
Evaluation Metrics	Accuracy, Recall, F1 Score, ROC-AUC

NEW NETWORK ARCHITECTURES

```
class DepthwiseSeparableBlock(nn.Module):  
  
    def __init__(self, in_channels: int, out_channels: int) -> None:  
        super().__init__()  
        self.block: nn.Sequential = nn.Sequential(  
            # 1. DEPTHWISE STEP: Spatial filtering (3x3)  
            # groups=in_channels forces each channel to be filtered independently  
            nn.Conv2d(in_channels, in_channels, kernel_size=3, padding=1, groups=in_channels, bias=False),  
            nn.BatchNorm2d(in_channels),  
            nn.LeakyReLU(),  
  
            # 2. POINTWISE STEP: Channel mixing (The 1x1 Point After)  
            # This blends the filtered channels together into the new channel size  
            nn.Conv2d(in_channels, out_channels, kernel_size=1, bias=False),  
            nn.BatchNorm2d(out_channels),  
            nn.LeakyReLU(),  
            nn.MaxPool2d(kernel_size=2)  
        )
```



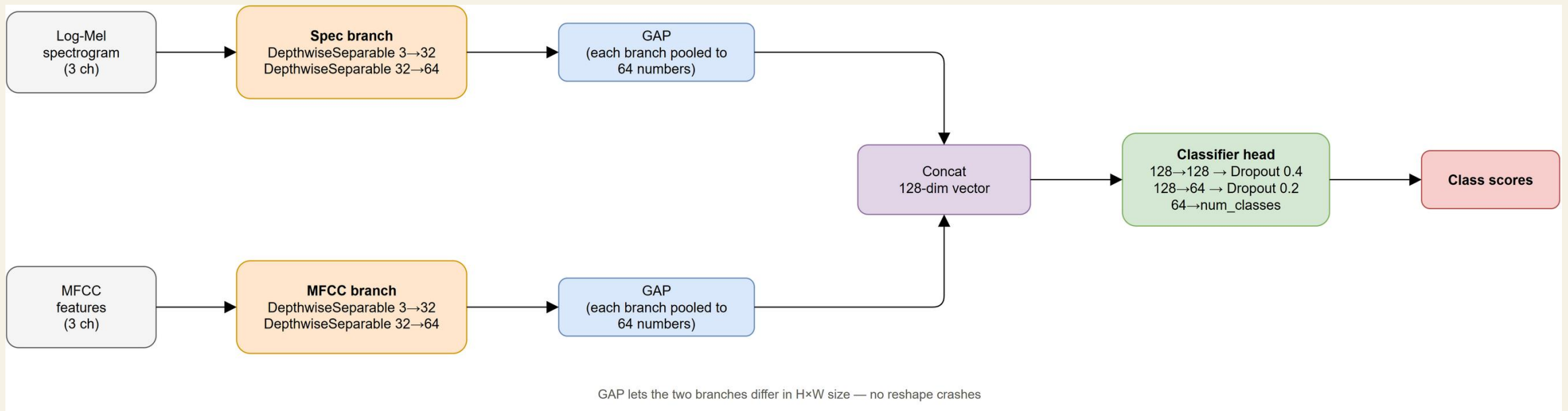
each step: BatchNorm → LeakyReLU

Normal 3x3 conv (3→32): 864 weights
Ours: depthwise 27 + pointwise 96 = 123 (~7× fewer)
64→128: 73,728 vs 8,768 (~8× fewer)

fewer params → less capacity to memorize
(MobileNet trick: split, don't pay for both jobs at once)

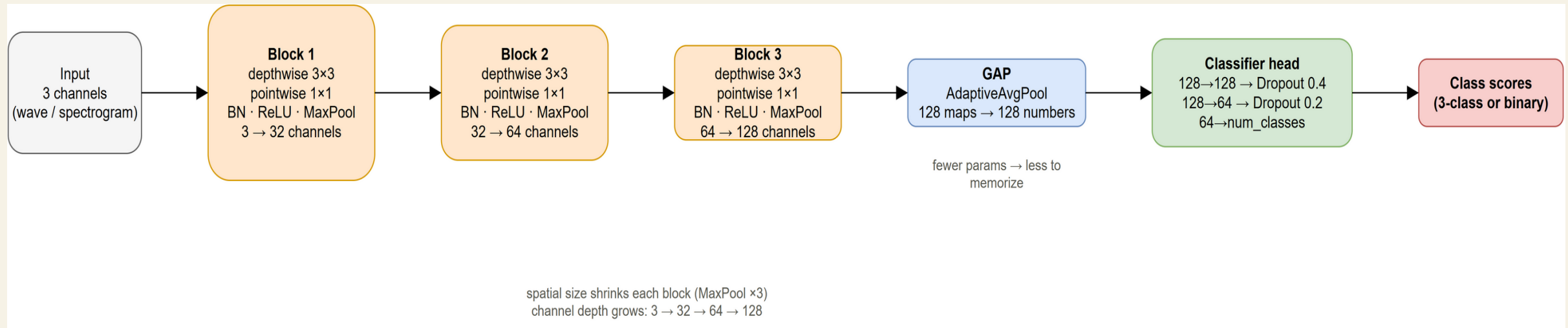
NEW NETWORK ARCHITECTURES

Branched Network



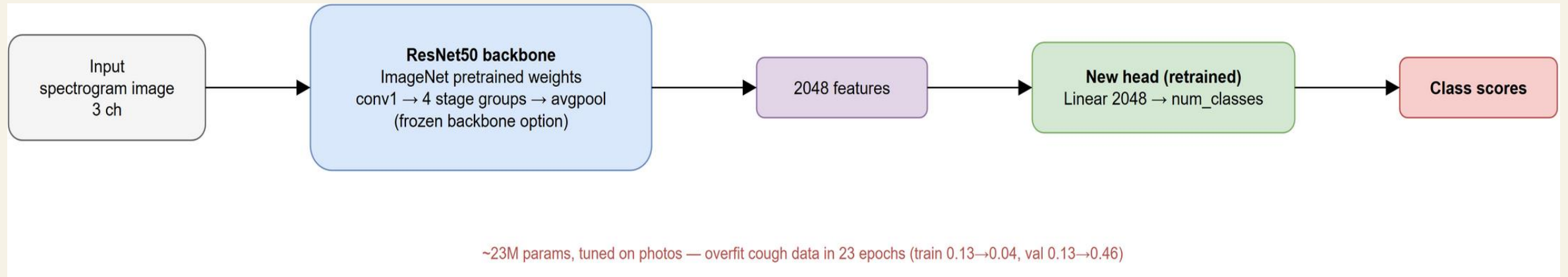
NEW NETWORK ARCHITECTURES

DefaultCNN Network



NEW NETWORK ARCHITECTURES

Resnet Model Network



THE 3-CLASS MODEL COLLAPSED

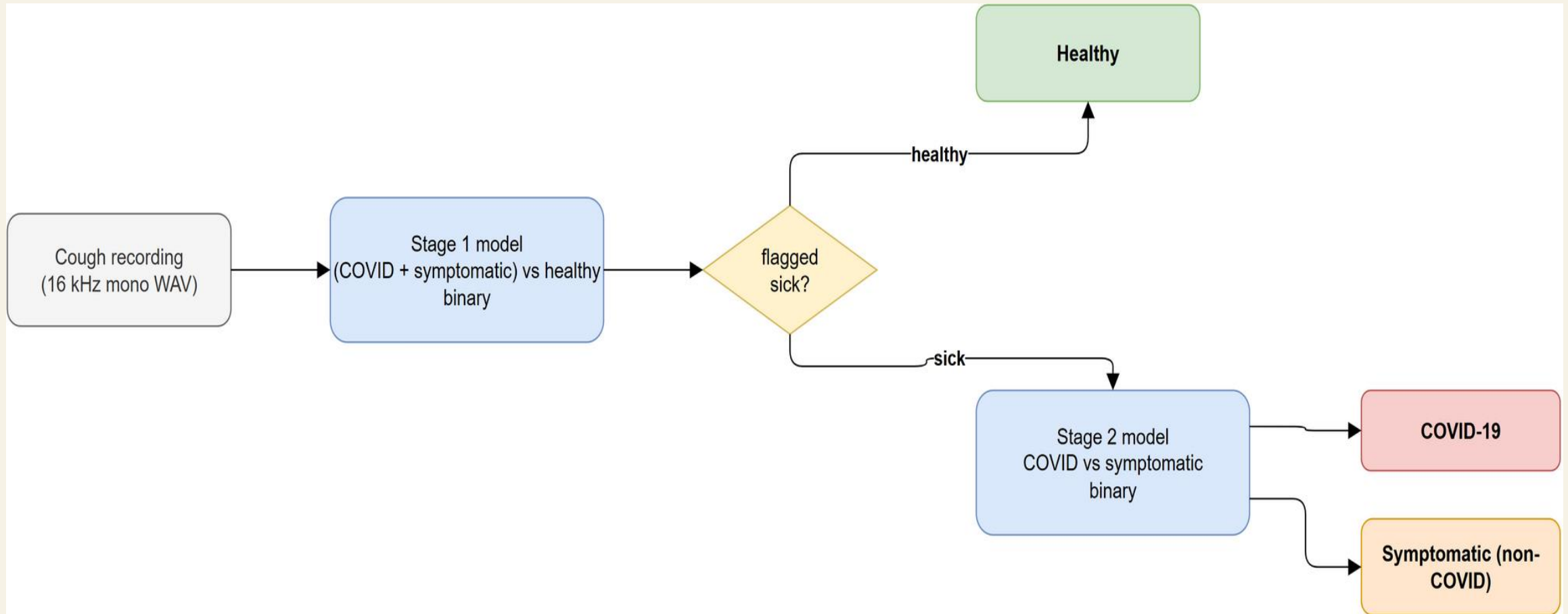
Problem:

- one 3-class CNN (healthy / symptomatic / COVID) on MFCC + log-Mel features, three input variants (wave-only, spec-only, both).

Every variant collapsed to the majority class:

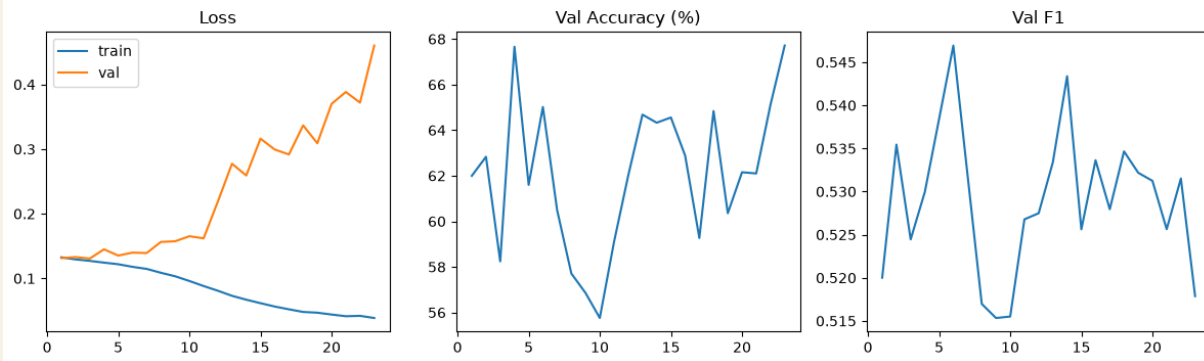
- Validation F1 frozen at **0.2459** for all 60 epochs
- Validation Accuracy stuck at ****58.43%**** while training loss fell and validation loss did not → memorizing, not learning

CASCADE OF MODELS SOLUTION

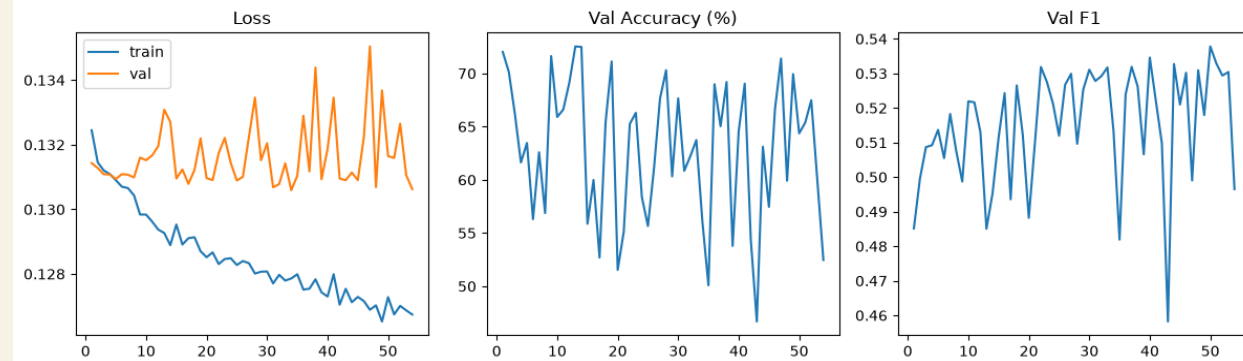


STAGE 1 RESULTS

trained resnet / covid_symp_vs_healthy / both

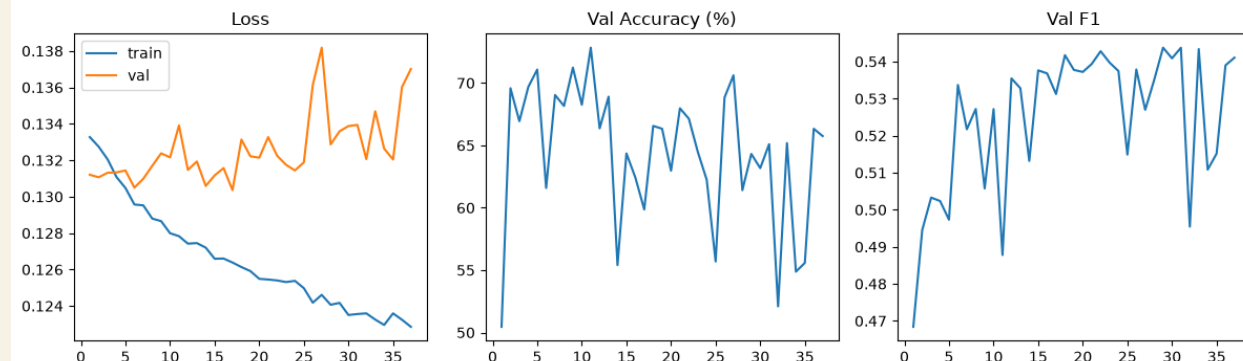


trained branched / covid_symp_vs_healthy / both

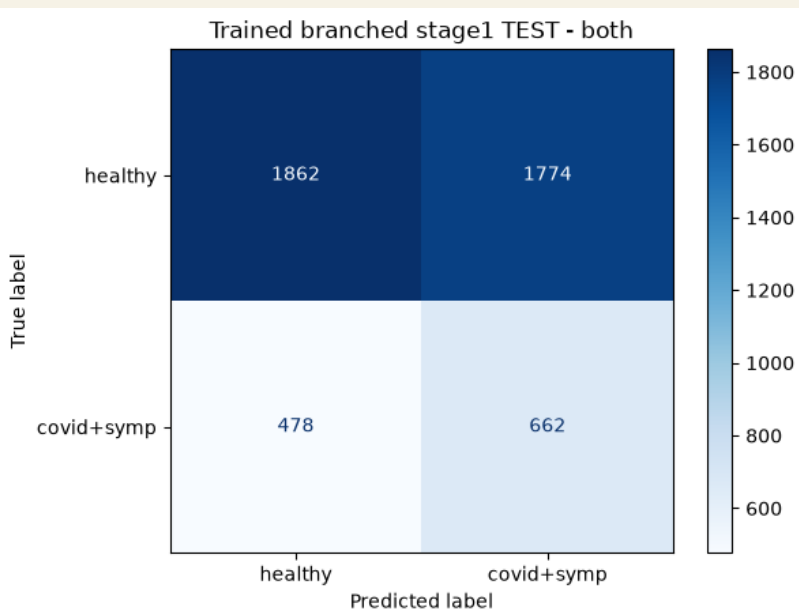
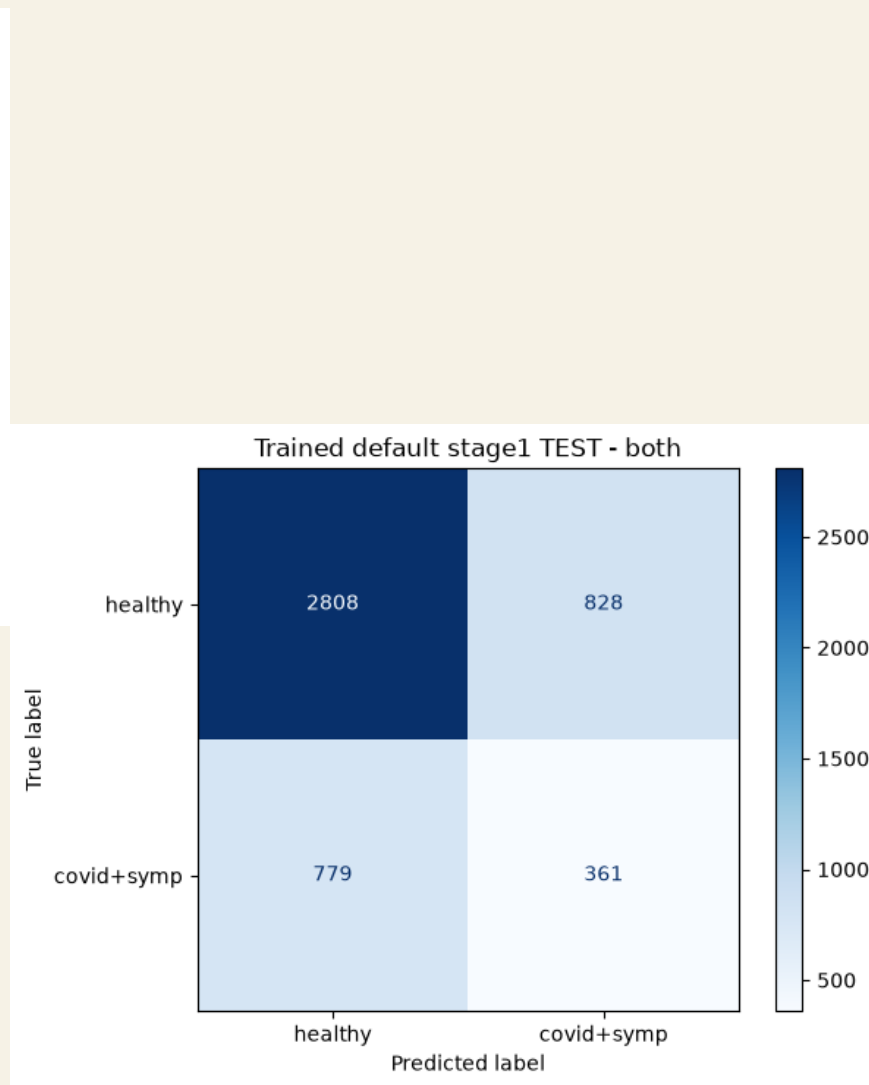
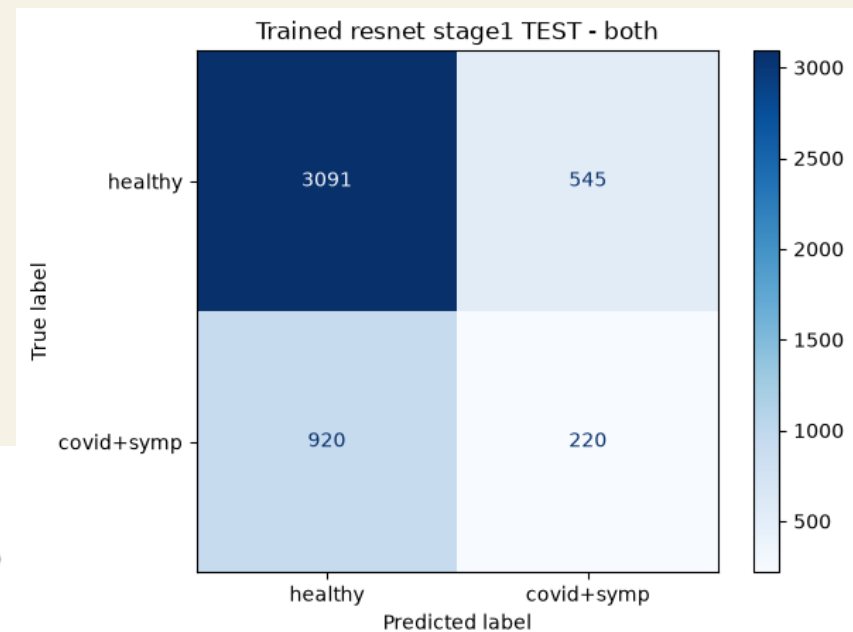


Model	Val AUC	Epochs
Default	0.57	37 (early stop)
Branched	0.56	54
ResNet50	—	23 (overfit, early-stopped)

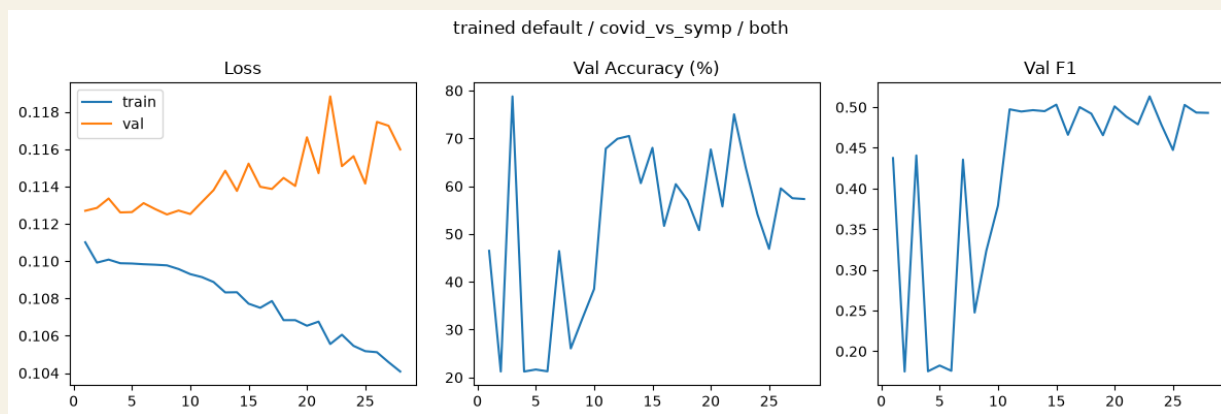
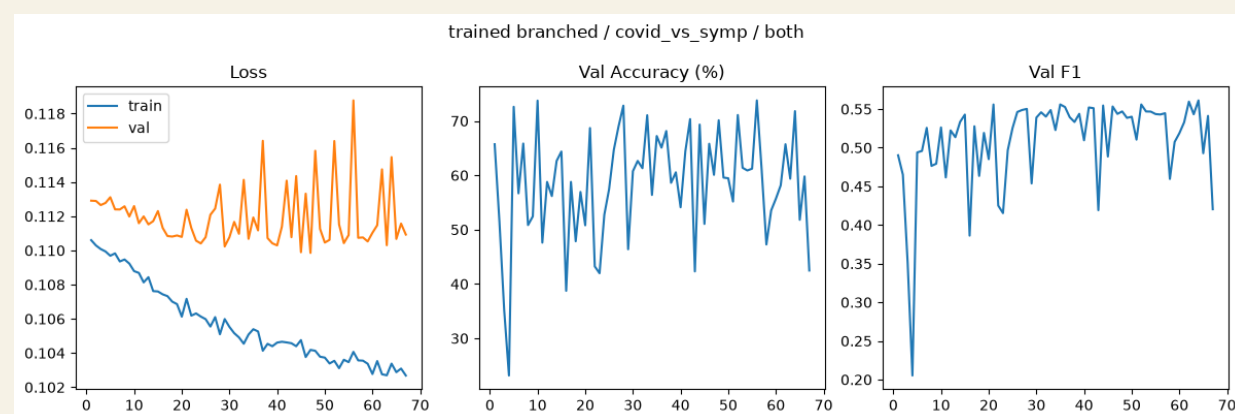
trained default / covid_symp_vs_healthy / both



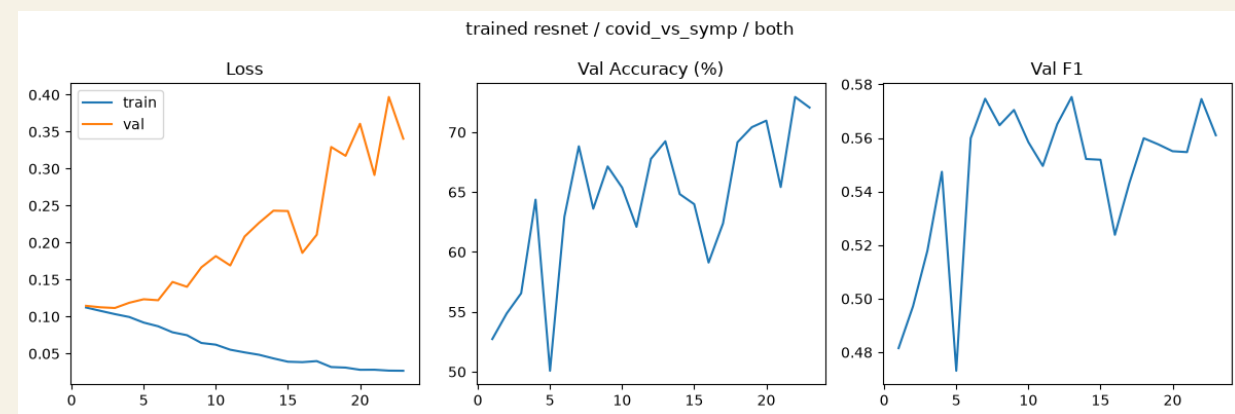
STAGE 1 CONFUSION MATRICES



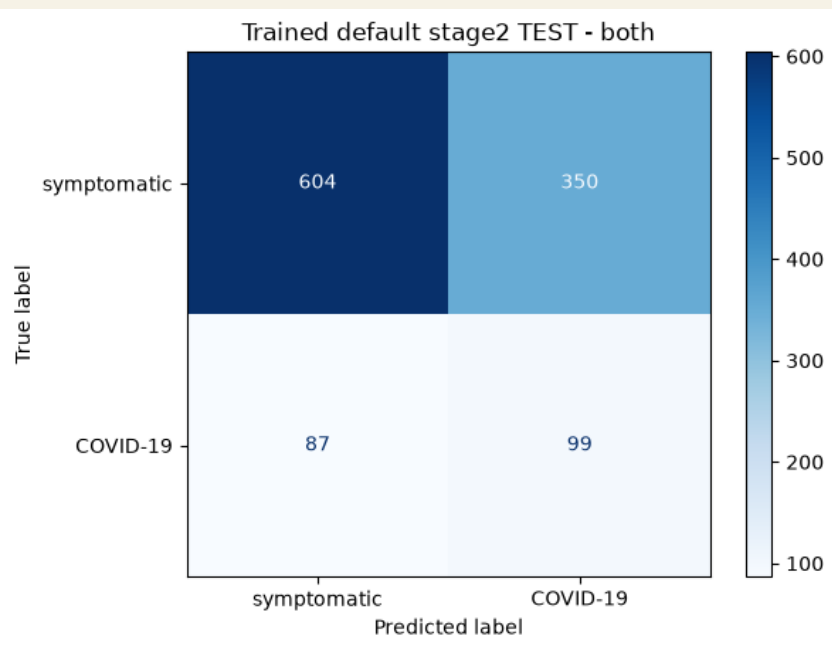
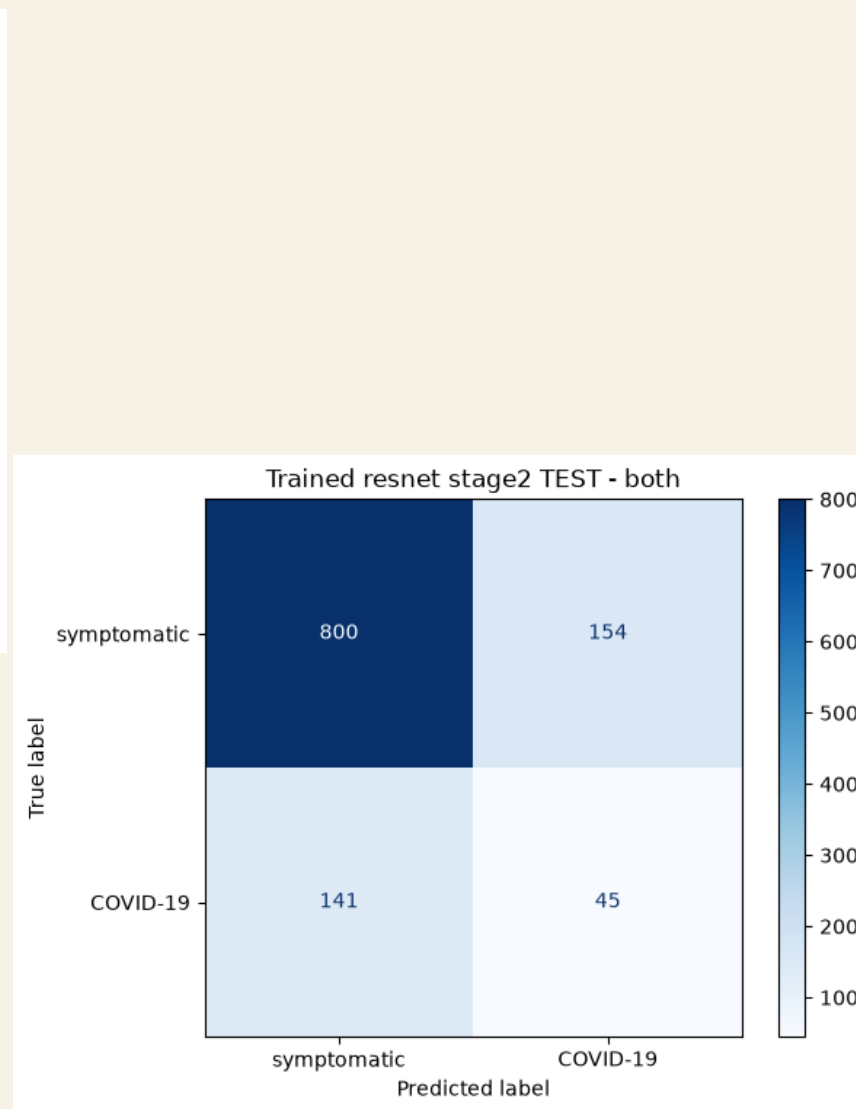
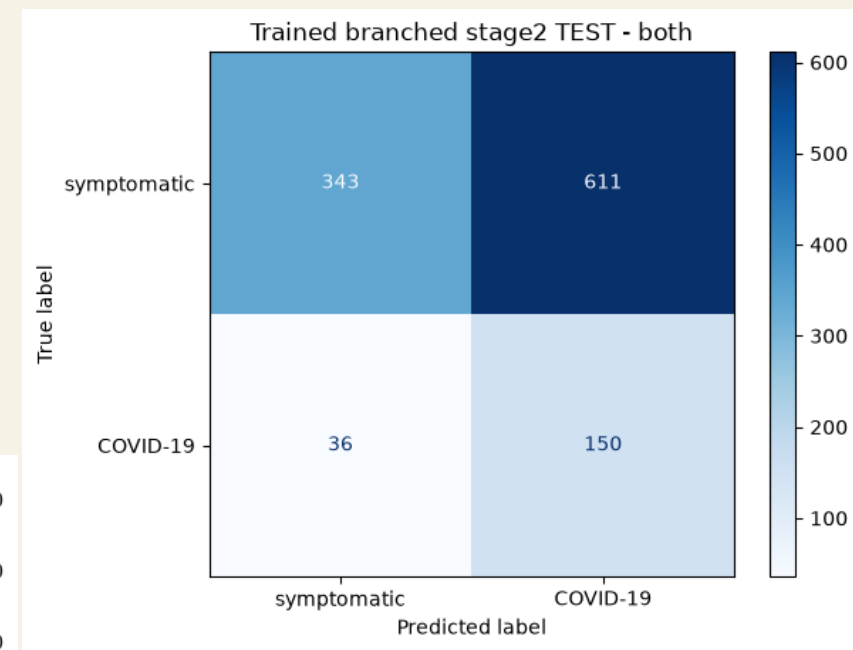
STAGE 2 RESULTS



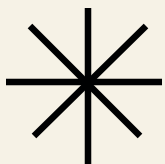
Model	Val AUC	Test AUC	COVID Recall
Default	0.532	0.599	0.47
Branched	0.623	0.637	0.61
ResNet50	0.627	0.584	0.5



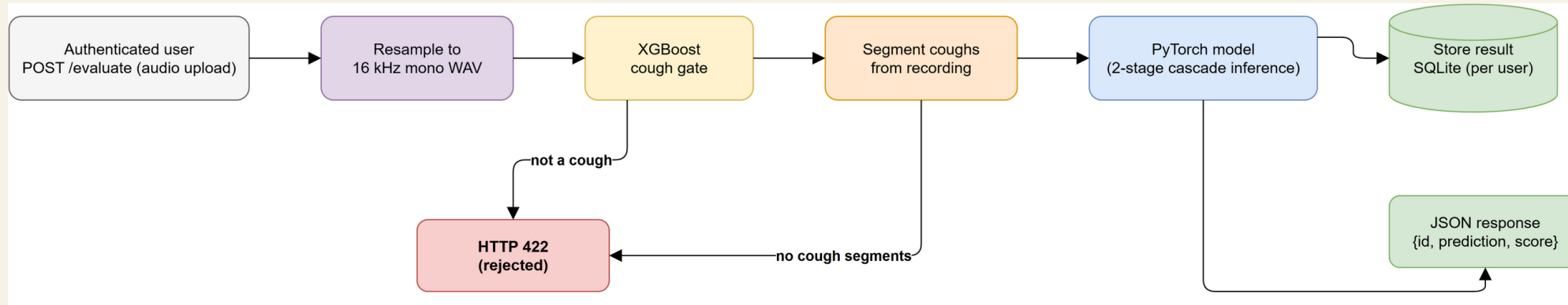
STAGE 2 CONFUSION MATRICES



BACKEND &
FRONTEND
DEVELOPMENT.



FASTAPI BACKEND



```
@router.post("/evaluate")
def evaluate(file: UploadFile = File(...), user=Depends(get_current_user)):
    raw = file.file.read()
    if not raw:
        raise HTTPException(400, "Empty upload")
    wav = to_wav_16k(raw)

    xgb_result = evaluate_xgboost(wav)
    if not is_success(xgb_result):
        raise HTTPException(422, "XGBoost gate rejected the sample")

    segments = segment_audio(wav)
    if not len(segments):
        raise HTTPException(422, "XGBoost gate rejected the sample")

    result = run_torch_model(segments)

    metadata = json.dumps({"filename": file.filename, "xgb_score": xgb_result["score"]})
    conn = get_conn()
    cur = conn.execute(
        "INSERT INTO evaluations (user_id, prediction, score, xgb_gate_success, metadata) VALUES (?, ?, ?, ?, ?)",
        (user["id"], result["prediction"], result["score"], 1, metadata),
    )
    conn.commit()
    return {"id": cur.lastrowid, "prediction": result["prediction"], "score": result["score"]}
```

DEMO



[DEMO LINK](#)

WHAT'S NEXT

- Evaluate and integrate additional data sources to grow the training set
 - Look for new data sources and integrate them into our existing dataset
- Improve preprocessing and feature extraction for the cascade models
 - Ensure speaker leakage does not occur
 - Find new audio processes for new features
- Rebalance the training data and use class-weighting strategy
 - Try and keep class proportions per data set and batch the same
- Refine stage 1 of the cascade: thresholds, architecture, and sick-class recall
 - Try to improve stage 1 of our cascade to reliably predict if someone is healthy or sick
 - May use a different architecture for stage one (like Wav2Vec)
- Retrain and re-evaluate the full cascade at the improved operating point
 - Training on lower learning rates and evaluating full pipeline
 - Train all new models

MILESTONE 3 TASK MATRIX

Task Description	Assigned Team Member	Status
Evaluate and integrate additional data sources to grow the training set	All	In Progress
Improve preprocessing and feature extraction for the cascade models	All	In Progress
Rebalance the training data and revisit the class-weighting strategy	All	In Progress
Refine stage 1 of the cascade: thresholds, architecture, and sick-class recall	All	Pending
Retrain and re-evaluate the full cascade at the improved operating point	All	Pending

QUESTIONS??

