DELEGATE BOOKLET : Guidelines and competition framework

01| What is the BioFusion Hackathon?

The BioFusion Hackathon is a medical-themed AI/ML competition held in conjunction with the BioFusion Science Convention. Conducted entirely online, the hackathon invites participants to develop innovative machine-learning solutions to a healthcare or medical problem of their choice. It serves as a platform for students to apply their technical skills to real-world challenges, showcase creativity, and contribute to the future of medical technology.

02| Event Structure

Teams choose one medical/ healthcare related problem and propose an innovative, machine learning solution using datasets available on an open source dataset repository of their choice.

Teams build a working machine learning model focusing on creativity, relevance, and feasibility.

Teams submit their Google Colab notebooks and report via the provided Google Form link

Submissions are evaluated by a panel of subject expert academics, and finalists are announced.

03| Event Overview

Participants design and implement a complete solution using AI for a healthcare or medical-related problem of their choice.

Each team will submit:

1. A Google Colab Notebook (with full code with comments, logs, outputs, metrics).

2. A Technical + Application Report (PDF).

Finalists are selected based on technical evaluation, documentation quality, and model performance.

04| Project Requirements

Participants must select:

1. A medical/healthcare problem

2. A dataset from an open source dataset repository (eg: Kaggle, OpenML)

05| Allowed Resources

✔ **Allowed**

- Any dataset from an **open source** dataset repository.

- Jupyter Notebook, Google Colab, Kaggle Notebook.

- Publicly available Python packages (TensorFlow / PyTorch / scikit-learn / OpenCV / NumPy / Pandas / Matplotlib etc.).
- Pretrained models, pretrained weights and embeddings are allowed, but the details of the pretrained model and model architecture should be clearly documented and justification for choosing the specific model should be provided.

- Academic papers, online documentation, textbooks.

❌ **Not Allowed**

- Datasets from repositories that require subscriptions or membership are not allowed.

- Should not be another person's work. If proven even after announcing the winners teams can be disqualified.

- AutoML tools

- Previously built pipelines.

06| Submission Requirements

Submit the following renamed in the given format;

A. Notebook - renamed as **TeamName_Notebook** (File format : .ipynb)
B. Report - renamed as **TeamName_Report** (File format: .pdf)

A. Notebook (Required Format)

Your notebook **must include all of the following**:

1. Problem Definition

- Clinical/healthcare relevance

- What is being predicted?

2. Dataset Documentation

- Full citation of the dataset

- Variables / labels description

- Data distribution + basic analysis

- Preprocessing steps taken

3. Model Initialization & Pretraining Disclosure

A. Pretrained model used

- Model name
- Source
- Task it was originally trained on (e.g., ImageNet, clinical text, generic images)

B. Weight usage

Did you use existing learned weights or did you start from scratch?

You must clearly state one of these:

- Used pretrained weights.
- Random initialization (training from scratch)

4. Model Development

- Full architecture (from scratch) or using a pretrained model and justification for the chosen design.
- List and explain all hyperparameter choices (learning rate, batch size, optimizer, etc.) ● Fine-tuning / Training Strategy.
- Clearly show in code and markdown:
  - Which layers are frozen
  - Which layers are partially trainable
  - Which layers are fully trainable
  - Whether pretrained backbone is used only as a feature extractor or fully fine-tuned

Deep Learning Models

- You must **show how your model learns** by displaying the following steps:
  - Forward pass → how inputs produce predictions
  - Loss computation → how errors are calculated
  - Backpropagation → how gradients are computed
  - Optimizer update → how model weights are updated
- **Do not hide** training steps in one-line automated calls/ high level functions (eg: `model.fit()`)without explanation.

Machine Learning Models

- Clearly document your training procedure in code and markdown:
    - How the model was trained
    - Hyperparameters used
    - Data splitting strategy

Validation Approach

- Use train/validation/test split or k-fold cross-validation ● Explain your choice and show it in code/markdown

5. Outputs & Logs

- Training curves (loss, accuracy, etc.)
- Validation metrics
- Error analysis
- Computational constraints faced

6. Performance Metrics, including but not limited to

- Primary metrics
- Secondary metrics
- Confusion matrix (if classification)
- ROC curves (if applicable)

7. Reproducibility

- Code cells must run when configurations are set up
- Random seeds must be set

8. Final Model File

B. Report (PDF)

Page Limit: 5 pages

Font Size : 12

Font Type : Times New Roman

Line Spacing : 1.15

Your report must include:

1. Literature Review

- Min 3 research papers
- What has been done before?
- Gaps in existing work
- What your solution improves or proposes

2. Problem Identification

- Who is affected?
- Why is this problem important?
- Specific unmet need in healthcare

3. Dataset Justification - Why the selected dataset is appropriate

4. Methodology

- Data preprocessing pipeline
- Model architecture (diagrams encouraged)
- Training process
- Validation strategy

5. **Pretrained Model Usage & Adaptation** This

should include:

a. Rationale

- Why a pretrained model was chosen
- Why it is appropriate for the medical task

b. Modifications

- Architectural changes made (e.g., replaced classifier head)
- New layers added
- Output adaptation

c. Training strategy

- Fine-tuning vs feature extraction
- Learning rates used for:
  - Pretrained layers
  - Newly added layers (if different)

d. Risk & bias discussion

- Domain mismatch (e.g., natural images → medical images)
- Potential bias inherited from pretraining data

5. Results

- Metric tables
- Visualizations
- Error analysis
- Limitations of your model

6. Real-world Application

- Proposed deployment scenario
- Potential users (clinicians, patients, hospitals)
- Integrating into healthcare workflow
- Risks & limitations

7. Marketing & Impact Strategy

- Who would adopt it?
- Practical benefits
- Cost, accessibility, reach

8. Future Improvements

- Model enhancements

- Additional data needs

- Clinical translation pathways

07| Rules and Regulations

Rules About Data

✔ Must be from an open source database repository.
✔ Teams may combine multiple datasets
✔ Data set used must have > 300 records. When using smaller datasets, competitors must clearly document the justification for the use of a smaller dataset.
✔ Standard data augmentation techniques (e.g., rotation, flipping, cropping) are allowed.
✘ Adding samples from external datasets or scraping additional images/text is not allowed.

Use of pretrained components must be explicitly declared

Other Rules

- All submissions due by **8 PM** on 4th Jan 2026.

- Only 1 submission per team

- Late submissions will not be accepted

- Plagiarism or code copying leads to disqualification

08| Evaluation criteria

Projects will be evaluated based on methodological rigor, transparency, and appropriateness of design, regardless of whether models are trained from scratch or initialized with pretrained weights.

The judge's decision will be the final decision.

A. Technical Evaluation (Model + Methods) — 65%

Category	**Weight**	**Details**
Model design and adaptation	**20%**	Appropriateness of model choice for the medical problem. **If pretrained:** correctness of adaptation **If from scratch:** soundness and originality of architecture Providing clear justification of design decisions.
Data Understanding & Preprocessing	**10%**	Dataset suitability and understanding Handling of imbalance, noise, missing data Prevention of data leakage
Training & Validation Strategy	**10%**	Correct training loop implementation Hyperparameter choices and justification Validation method (train/val/test, k-fold)

Performance Metrics (task-appropriate)	**20%**	Correct choice of metrics for the task Honest reporting of results Interpretation of metrics Comparison with baselines (if provided)
Error Analysis and limitations	**5%**	Insight into failures, Model weaknesses

---

B. Notebook Quality — 20%

Category	**Weight**	**Details**
Code Clarity & Documentation	**10%**	Readability, comments, logical cell order
Reproducibility	**10%**	Runs without errors, paths intact, no hidden dependencies, Explicit disclosure of pretrained components.

C. Technical & Application Report — 15%

Section	Weight	Details
Literature Review & Problem Identification	**5%**	Understanding of clinical context + research foundation
Methodology Explanation	**5%**	Clear reasoning, workflow diagrams, decisions justified Transparency of pretrained model usage, Appropriateness of pretraining source Discussion of bias, domain mismatch, and limitations
Results, Discussion, Real-World Application	**5%**	Interpretation of findings, practicality, limitations