# Classification of Brain MRI Scans Using Teachable Machine

Andreu Massanet Felix
Diego Malagrida González

**Abstract**

This project presents the development and analysis of a brain tumor classification system from magnetic resonance imaging (MRI) using *transfer learning* techniques with Teachable Machine. A model was trained to distinguish among four categories (glioma, meningioma, pituitary, and no tumor), and its performance was evaluated using an independent test set. The study focuses on analyzing hyperparameters such as learning rate, batch size, and number of epochs, showing how these variables affect gradient stability. Finally, the visual biases detected in the images are discussed, and a functional web application is presented that enables real-time inference, highlighting the importance of human oversight in automated medical diagnosis.

# 1 Problem/Idea Definition and Justification

Automatic medical image classification is a field of great interest due to its potential to support clinical decision-making, speed up triage, and reduce workload. This project proposes a simplified academic scenario: training a model capable of identifying visual patterns associated with different types of brain tumors and with cases without tumors. The source code is available in a public repository[1] and includes a functional web application for running inferences[2]

## 1.1 Project Objective

The main objective is to train an image classification model using **Teachable Machine**, and then integrate it into a small application that allows:

- Uploading a static medical image.
- Obtaining a class prediction with probabilities.
- Applying a confidence threshold $\theta$ to detect uncertain predictions.

## 1.2 Motivation

1. **Realistic application:** tumor classification is a real task present in many research works.
2. **Transfer learning:** it allows leveraging pretrained models even with limited datasets.
3. **Critical analysis:** the medical domain is especially sensitive to errors, making it ideal to discuss limitations and biases.

[1] Project repository: https://github.com/diegoMalagrida/mri-tumor-classifier.
[2] Project web application: https://diegomalagrida.me/mri-tumor-classifier/app/.

## 2 Dataset Origin

The dataset used[3] comes from a public online source and contains brain medical images classified into four categories:

- **Glioma**
- **Meningioma**
- **Pituitary**
- **No tumor**

In addition, Figure 1 shows a representative sample of images from the dataset, where variability in slice selection, orientation, and contrast across patients can be observed.

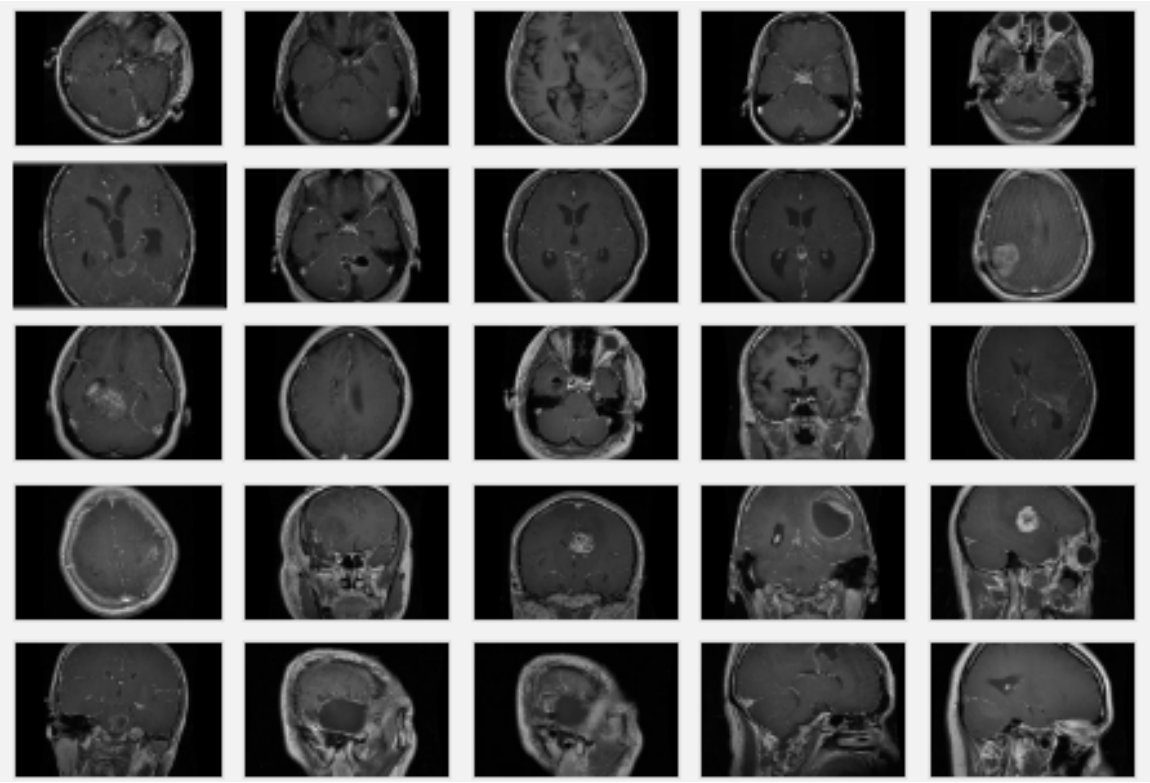


Figure 1: Dataset sample.

To facilitate training and evaluation without losing representativeness, we worked with a balanced subset of the dataset. Specifically, **500 images per class** were randomly selected from the *training* set. In addition, an independent test set of **100 images per class** was defined from the original *testing* split.

### 2.1 Subset Selection Criterion

Selection was performed using stratified random sampling, ensuring:

- **Randomness:** any manual selection was avoided to reduce bias.
- **Class balance:** all categories maintain the same number of examples.

[3] https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset

### 2.2 Image Size Normalization

In order to maintain a homogeneous input format for the model, a standard resolution of **512×512** was set, since it was the most frequent in the dataset.

However, it was observed that the **No tumor** class did not contain enough images at that resolution. To address this, the **500 largest available images** were selected and a uniform rescaling to **512×512** was applied using a Python script.

This normalization helps reduce format-related variability and prevents resolution from acting as a discriminative factor unrelated to medical content.

# 3 Model Training

## 3.1 Base Model and Transfer Learning

Teachable Machine uses a pretrained model as a feature extractor. Through **transfer learning**, the model learns a final classifier adapted to the 4 defined classes.

To facilitate training and evaluation without losing representativeness, we worked with a balanced subset of the dataset. Specifically, **500 images per class** were randomly selected from the *training* set, since preliminary tests with larger subsets significantly increased loading and training time. This is because Teachable Machine is mainly intended for quick experiments and moderately sized datasets, so 500 images per class was an appropriate compromise between representativeness and computational feasibility. In addition, an independent test set of **100 images per class** was defined from the original *testing* split.

Figure 2 shows the interface used, including the four classes and the advanced training configuration.

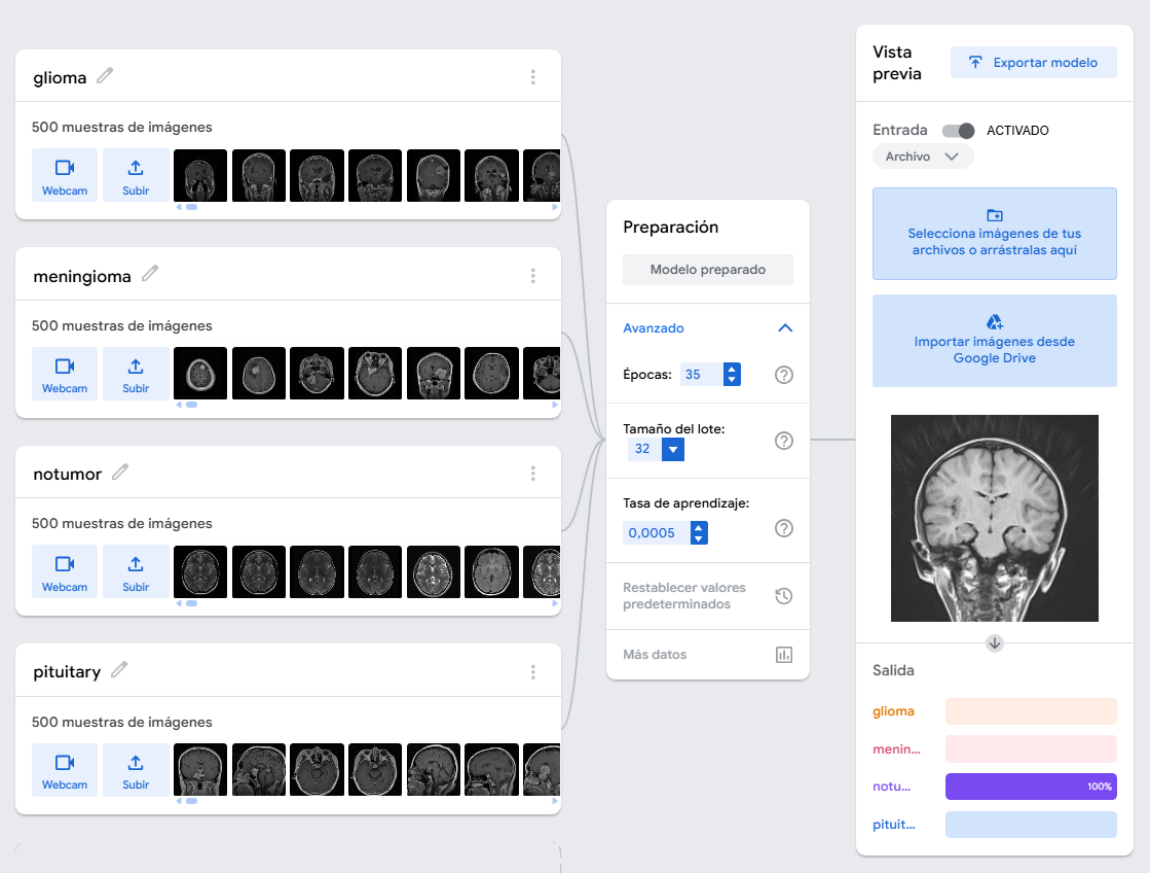



Figure 2: Teachable Machine interface showing class loading and training configuration using *transfer learning*.

# 4 Model Training and Main Configuration

## 4.1 Configuration Used

For training the final model, **transfer learning** was applied on a base architecture predefined by Teachable Machine. After a fine-tuning process, the following hyperparameters were selected as they offered the best trade-off between stability and accuracy:

- **Epochs:** 35
- **Batch size:** 32
- **Learning rate:** 0.0005

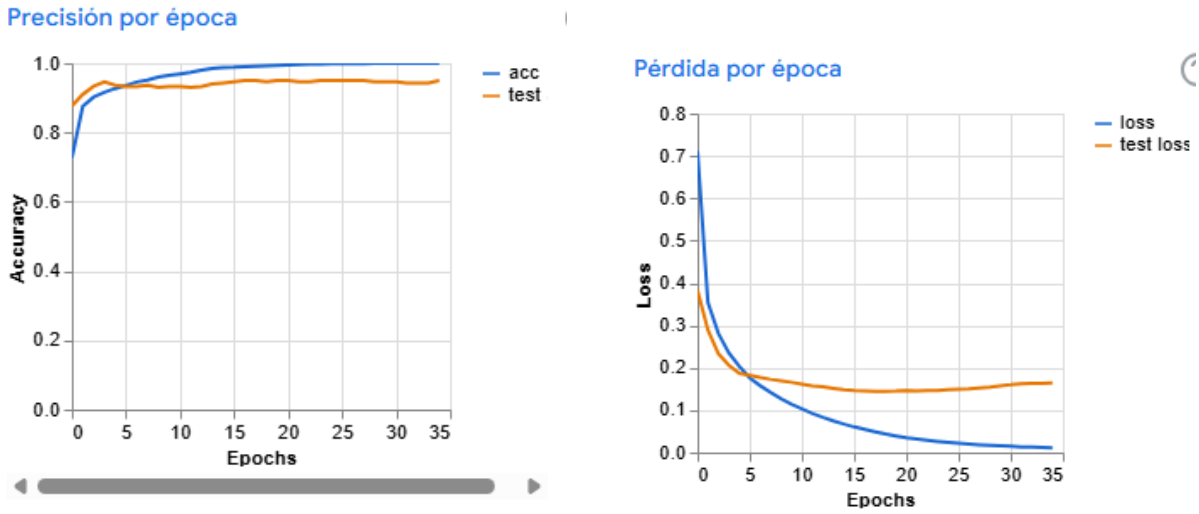
To reduce low-confidence incorrect predictions, a thresholding system was implemented:

$$\hat{y} = \begin{cases} \arg\max_k p(y = k \mid x) & \text{if } \max_k p(y = k \mid x) > \theta \\ \text{Unknown/uncertain class} & \text{otherwise} \end{cases}$$

In the final implementation, a default value of $\theta = 0.80$ was used (adjustable in the interface).

## 4.2 Training Results and Metrics

During training, learning progress was monitored through accuracy and loss curves. As shown in the following figures, the model reaches convergence around epoch 25, where the error stabilizes.




(a) Accuracy per epoch.

(b) Loss per epoch.

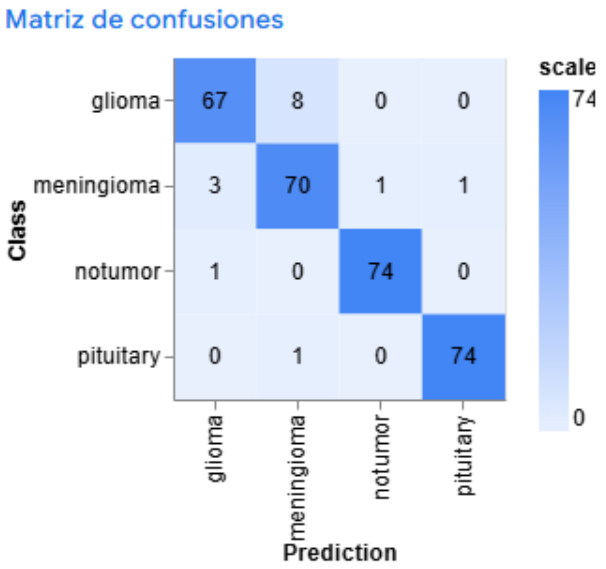
Figure 3: Training evolution.

Likewise, performance for each category was evaluated through class-specific accuracy and the confusion matrix, which helped identify which classes exhibit more similar morphologies for the classifier.

Precisión por clase

| CLASS | ACCURACY | # SAMPLES |
|---|---|---|
| glioma | 0.89 | 75 |
| meningioma | 0.93 | 75 |
| notumor | 0.99 | 75 |
| pituitary | 0.99 | 75 |

(a) Accuracy per class.




(b) Confusion matrix.

Figure 4: Detailed metrics for the main configuration.

### 4.3 Confusion Matrix

Figure 5 shows the model confusion matrix on the test set. In general, the dominant diagonal reflects a good number of correct predictions, especially for the *notumor* (96/100) and *pituitary* (96/100) classes.

The most frequent errors occur between *glioma* and *meningioma*, indicating that the model tends to confuse both tumor subtypes (14 *glioma* cases predicted as *meningioma* and 2 in the opposite direction).

However, in a clinical setting the most relevant error is not confusing the *type* of tumor, but **failing to detect its presence**. In that sense, the most critical failure corresponds to predicting *notumor* when the image belongs to a tumor class (false negative). This pattern is observed mainly in *meningioma* → *notumor* (16 cases), suggesting that some meningiomas have less evident features or are more similar to images without tumors.

Confusion Matrix (Actual vs Predicted)

| Actual \ Predicted | Glioma | Meningioma | Notumor | Pituitary |
|---|---|---|---|---|
| Glioma | **85** | 14 | 0 | 1 |
| Meningioma | 2 | **70** | 16 | 12 |
| Notumor | 3 | 0 | **96** | 1 |
| Pituitary | 0 | 3 | 1 | **96** |

Figure 5: Model confusion matrix on the test set.

### 4.4 Classification Report

Figure 6 summarizes model performance using *precision*, *recall*, and *F1-score* for each class.

Overall, the model achieves **87% accuracy**. The best-performing classes are *notumor* and *pituitary* (F1-score ≈ 0.90), showing consistent behavior in both precision and sensitivity. In contrast, *meningioma* has the lowest performance (F1-score 0.75), suggesting greater visual overlap with other categories and aligning with what is observed in the confusion matrix.

Finally, since the test set is balanced (*support*=100 per class), the *macro* and *weighted* averages are very similar, reflecting overall performance without class imbalance bias.

**Classification Report**

| Metric \ Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Glioma** | 0.94 | 0.85 | 0.89 | 100 |
| **Meningioma** | 0.80 | 0.70 | 0.75 | 100 |
| **Notumor** | 0.85 | 0.96 | 0.90 | 100 |
| **Pituitary** | 0.87 | 0.96 | 0.91 | 100 |
| Accuracy | | | **0.87** | 400 |
| **Macro Avg** | 0.87 | 0.87 | 0.86 | 400 |
| **Weighted Avg** | 0.87 | 0.87 | 0.86 | 400 |

Figure 6: *Classification report* of the model on the test set.

### 4.5 Comparison: Internal vs. External Evaluation

When comparing results, it can be seen that Teachable Machine's internal evaluation provides extremely optimistic metrics, reaching accuracies of up to 99% in some classes (Figure 4b). This discrepancy occurs because its internal validation uses a subset of images that share the same style, resolution (512×512), and acquisition conditions as the training data.

In contrast, external validation on the independent test set reduces overall accuracy to 87% (Figure 6). In this scenario, the model faces images with greater variability in format and sharpness that it has never seen before. This performance gap confirms slight *overfitting* to the visual style of the training dataset and demonstrates that, in a real medical environment, generalization capability is lower than what rapid training tools may suggest.

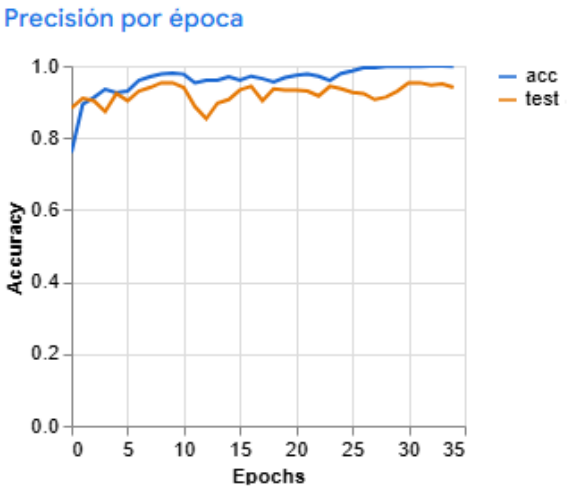
# 5 Experimentation with Other Hyperparameters

To better understand the learning dynamics of the model, tests were performed by modifying optimization parameters. These tests make it possible to observe how the network reacts to extreme configurations.

## 5.1 Case 1: High Learning Rate

Epochs: 35 | Batch size: 32 | **Learning Rate: 0.01**

When using an excessively high learning rate, the optimizer takes steps that are too large, preventing the model from settling into a minimum of the loss function. This is reflected in the loss-per-epoch graph, which shows abrupt oscillations and instability spikes. Although training accuracy may appear to scale quickly, the divergence between training and validation curves suggests that the model is not extracting useful features, but rather bouncing erratically between possible solutions.



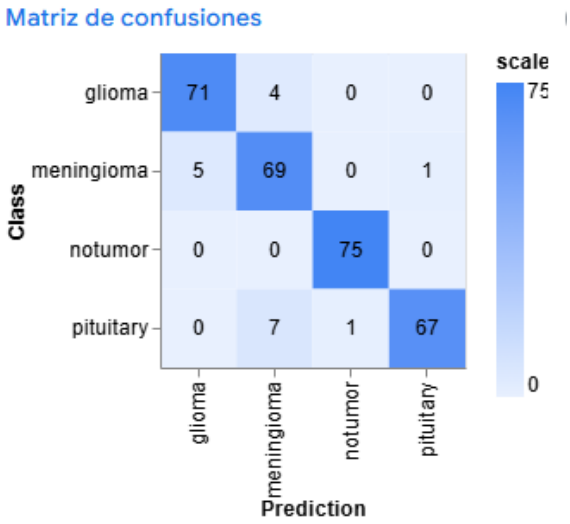

(a) Accuracy per epoch.


(b) Loss per epoch.

Precisión por clase

| CLASS | ACCURACY | # SAMPLES |
|---|---|---|
| glioma | 0.95 | 75 |
| meningioma | 0.92 | 75 |
| notumor | 1.00 | 75 |
| pituitary | 0.89 | 75 |

(c) Accuracy per class.




(d) Confusion matrix (Training).

Figure 7: Training results for Case 1.

Despite the instability observed during training, the model reaches an overall accuracy of 86% on the independent test set, slightly outperforming the other test cases. However, the confusion matrix reveals a critical vulnerability in identifying *meningioma*, where 21 cases are incorrectly classified as healthy brains (*notumor*). This level of false negatives is especially dangerous in a clinical context, as it implies missing an existing diagnosis. Although the F1-score for classes such as *pituitary* is high (0.93), the lack of consistency derived from an aggressive *learning rate* makes the model unreliable for safe medical deployment.

**Classification Report**

| Metric \ Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Glioma** | 0.88 | 0.88 | 0.88 | 100 |
| **Meningioma** | 0.82 | 0.61 | 0.70 | 100 |
| **Notumor** | 0.82 | 0.98 | 0.89 | 100 |
| **Pituitary** | 0.91 | 0.96 | 0.93 | 100 |
| Accuracy | | | **0.86** | 400 |
| **Macro Avg** | 0.86 | 0.86 | 0.85 | 400 |
| **Weighted Avg** | 0.86 | 0.86 | 0.85 | 400 |

**Confusion Matrix (Actual vs Predicted)**

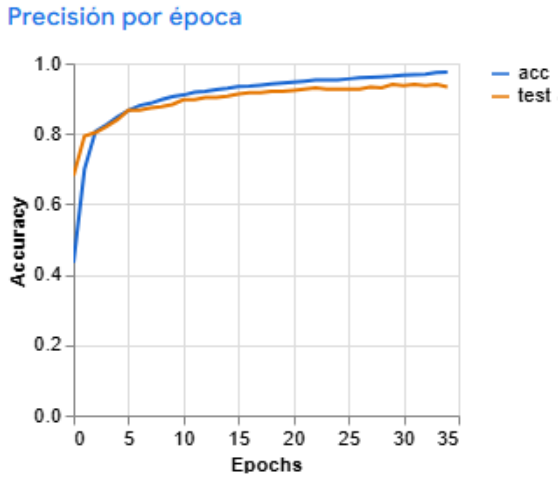
| Actual \ Predicted | Glioma | Meningioma | Notumor | Pituitary |
|---|---|---|---|---|
| **Glioma** | **88** | 11 | 0 | 1 |
| **Meningioma** | 9 | **61** | 21 | 9 |
| **Notumor** | 2 | 0 | **98** | 0 |
| **Pituitary** | 1 | 2 | 1 | **96** |

Figure 8: External validation for Case 1.

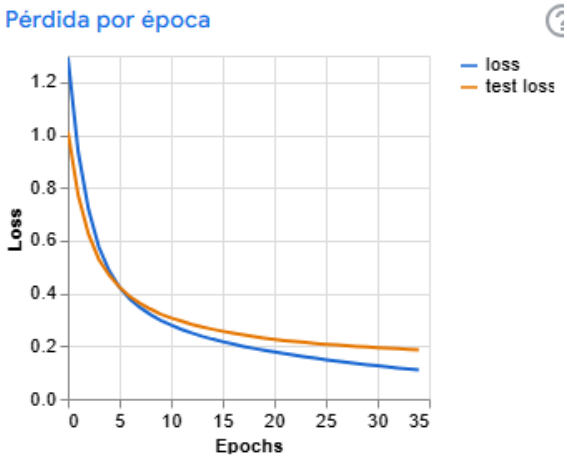
## 5.2 Case 2: Excessively Large Batch Size

Epochs: 35 | **Batch size: 512** | Learning Rate: 0.0005

When using the maximum batch size, the model performs very few weight updates per epoch, resulting in an extremely stable but slow learning process. The training graphs show a very flat and smoothed learning curve, without the typical oscillations of smaller batches. This behavior tends to over-average visual characteristics, which leads to increased difficulty in distinguishing between classes with similar morphologies, as seen in the confusions between glioma and meningioma in the results matrix.




(a) Accuracy per epoch.

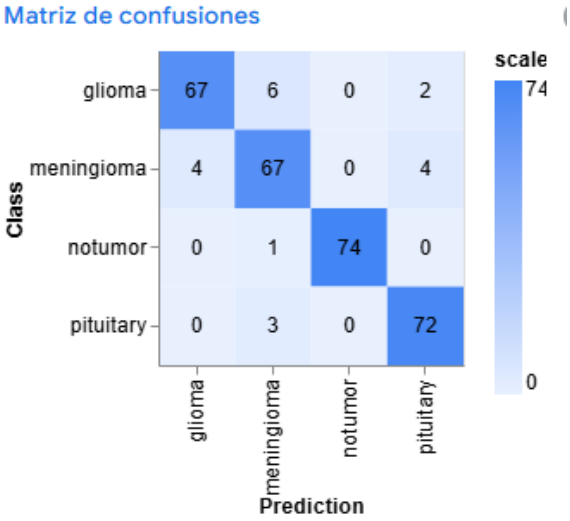



(b) Loss per epoch.

Precisión por clase

| CLASS | ACCURACY | # SAMPLES |
|---|---|---|
| glioma | 0.89 | 75 |
| meningioma | 0.89 | 75 |
| notumor | 0.99 | 75 |
| pituitary | 0.96 | 75 |

(c) Accuracy per class.




(d) Confusion matrix (Training).

Figure 9: Training results for Case 2.

External validation yields an overall accuracy of 84%, slightly higher than the low-epoch case but still below the optimal model. When analyzing the confusion matrix, it is observed that the model remains highly reliable at identifying healthy brains (*notumor*), but shows significant difficulties in the differential classification of tumors. Notably, *meningioma* is repeatedly confused with *pituitary* (19 cases) and *glioma* (6 cases), resulting in an F1-score of only 0.70 for this category. This confirms that an excessively large batch size prevents the optimizer from adjusting weights with the precision required to separate morphologically similar classes.

**Classification Report**

| Metric \ Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Glioma** | 0.91 | 0.86 | 0.88 | 100 |
| **Meningioma** | 0.77 | 0.64 | 0.70 | 100 |
| **Notumor** | 0.89 | 0.95 | 0.92 | 100 |
| **Pituitary** | 0.80 | 0.92 | 0.86 | 100 |
| Accuracy | | | **0.84** | 400 |
| **Macro Avg** | 0.84 | 0.84 | 0.84 | 400 |
| **Weighted Avg** | 0.84 | 0.84 | 0.84 | 400 |

**Confusion Matrix (Actual vs Predicted)**

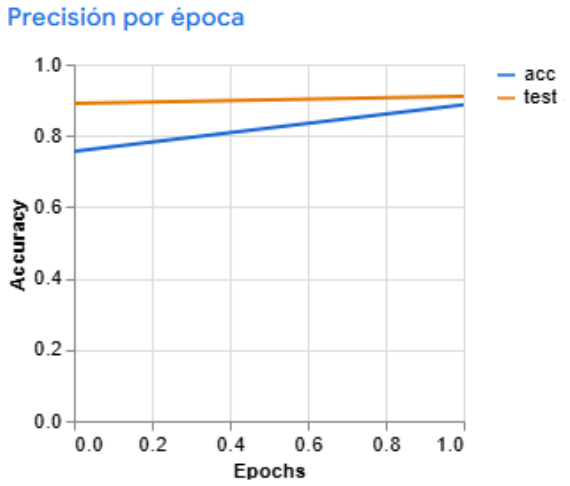
| Actual \ Predicted | Glioma | Meningioma | Notumor | Pituitary |
|---|---|---|---|---|
| **Glioma** | **86** | 13 | 0 | 1 |
| **Meningioma** | 6 | **64** | 11 | 19 |
| **Notumor** | 2 | 0 | **95** | 3 |
| **Pituitary** | 1 | 6 | 1 | **92** |

Figure 10: External validation for Case 2.

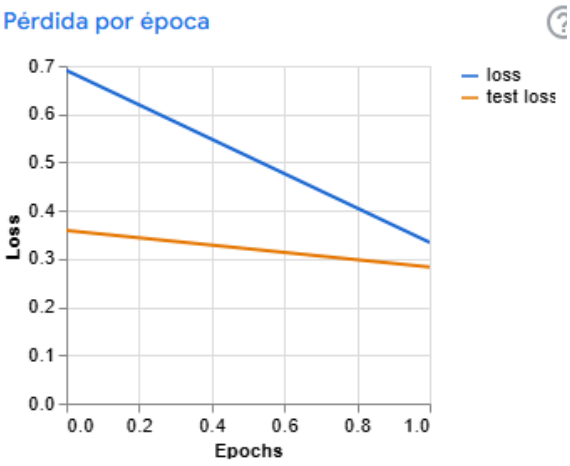
### 5.3 Case 3: Insufficient Training

**Epochs: 2** | Batch size: 32 | Learning Rate: 0.0005

In this scenario, the number of epochs was drastically limited to simulate an *underfitting* or under-training state. The evolution graphs show linear trajectories that do not stabilize, indicating that the model has not had enough iterations to adjust the weights of the final *transfer learning* layer. Although accuracy on the "notumor" class remains high, the system shows low and random reliability when distinguishing among different types of brain tumors due to insufficient training.




(a) Accuracy per epoch.

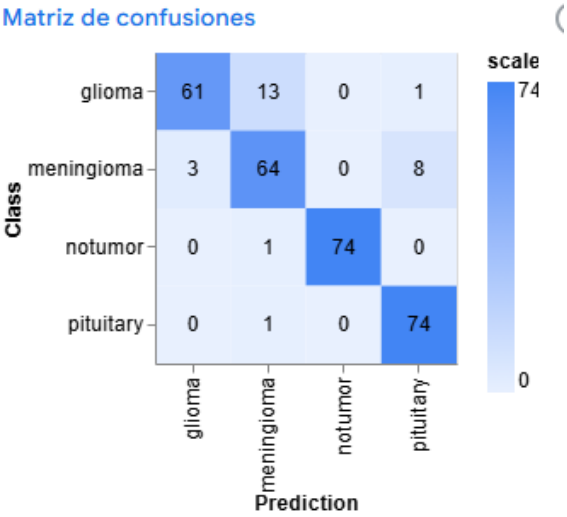



(b) Loss per epoch.

Precisión por clase

| CLASS | ACCURACY | # SAMPLES |
|---|---|---|
| glioma | 0.81 | 75 |
| meningioma | 0.85 | 75 |
| notumor | 0.99 | 75 |
| pituitary | 0.99 | 75 |

(c) Accuracy per class.




(d) Confusion matrix (Training).

Figure 11: Training results for Case 3.

Despite the brief training, the model reaches an overall accuracy of 82% on the independent test set. This seemingly high value is mainly explained by the robustness of the base model used in *transfer learning*. However, analysis of the confusion matrix reveals critical deficiencies in diagnostic reliability, especially in distinguishing between *glioma* and *meningioma*. The network shows a high volume of false negatives in the *meningioma* class, where a significant portion of samples are incorrectly classified as *pituitary*. These results demonstrate that, even if *accuracy* appears acceptable, two epochs are insufficient for the model to learn the subtle morphological features needed to safely differentiate tumor types, compromising its usefulness in a real clinical environment.

**Classification Report**

| Metric \ Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Glioma** | 0.92 | 0.78 | 0.84 | 100 |
| **Meningioma** | 0.73 | 0.62 | 0.67 | 100 |
| **Notumor** | 0.87 | 0.94 | 0.90 | 100 |
| **Pituitary** | 0.78 | 0.95 | 0.86 | 100 |
| Accuracy | | | **0.82** | 400 |
| **Macro Avg** | 0.82 | 0.82 | 0.82 | 400 |
| **Weighted Avg** | 0.82 | 0.82 | 0.82 | 400 |

**Confusion Matrix (Actual vs Predicted)**

| Actual \ Predicted | Glioma | Meningioma | Notumor | Pituitary |
|---|---|---|---|---|
| **Glioma** | **78** | 20 | 0 | 2 |
| **Meningioma** | 4 | **62** | 12 | 22 |
| **Notumor** | 2 | 1 | **94** | 3 |
| **Pituitary** | 1 | 2 | 2 | **95** |

Figure 12: External validation for Case 3.

# 6 Discussion of Biases and Errors

This section critically analyzes the model limitations, based on the discrepancy between the obtained metrics and the visual morphology of the processed medical images.

## 6.1 Visual Differences and Model Behavior

To understand systematic errors, it is necessary to analyze how the model interprets the visual and anatomical features of each class:

- **Pituitary (Pituitaria):** It shows the most stable behavior with an *F1-score* of 0.91. These tumors are always located in an anatomically invariant area (base of the brain), which makes detection easier.
- **Glioma:** It is characterized by being diffuse and altering the internal symmetry of the tissue. Its variability explains confusions with meningioma (14 cases).
- **Meningioma:** It appears as a compact, well-delimited mass that grows from the outer membranes inward. Despite its clarity, it is the class with the highest error rate.

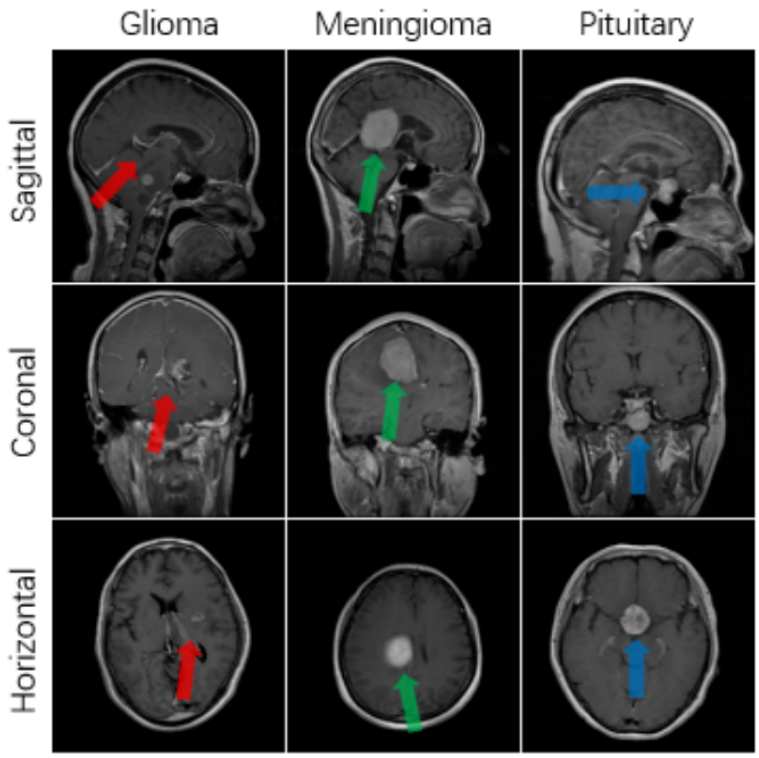



Figure 13: Visual differences among tumors.

## 6.2 Framing Bias and the Superior View Perspective

It is paradoxical that **Meningioma**, often presenting large protrusions, is the class with the most false negatives (16 cases classified as *Notumor*). After a visual analysis of the dataset, a critical bias was detected:

- **Prevalence of the superior plane:** The vast majority of images in the **Notumor** class are taken from a horizontal or superior plane where the scan occupies most of the frame.

- **Spurious correlation:** By chance, meningioma cases that the model incorrectly labels as healthy follow the same visual pattern: top-down views where the brain fills almost the entire frame.
- **Cranial mimicry:** From this perspective, a bump attached to the brain edge is visually confused with the natural curvature or thickness of the skull. The model prioritizes image composition (superior plane + high zoom) over the presence of the tumor mass, assuming that structure belongs to the normal anatomy of a healthy patient.

### 6.3 Failure Case Analysis (Qualitative Example)

To illustrate this limitation, a case is presented where image composition induces error. Figure 14 shows the MRI used as input; Figure 15 presents the classification summary; and Figure 16 shows the probability distribution.

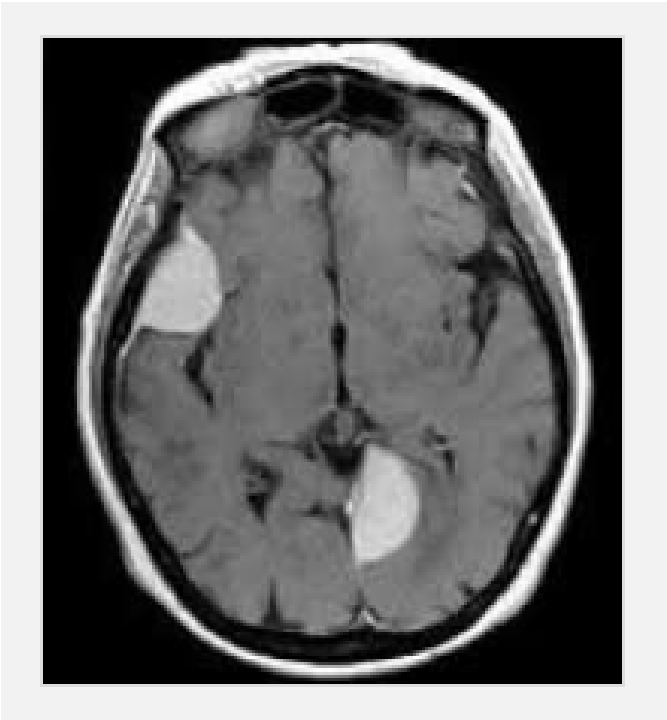


Figure 14: Critical failure example: input image with a top-down framing and high zoom.

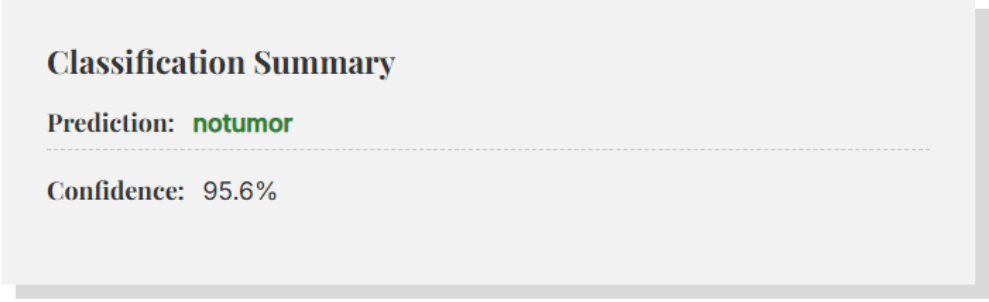



Figure 15: Case classification summary: the model predicts *Notumor* with high confidence.

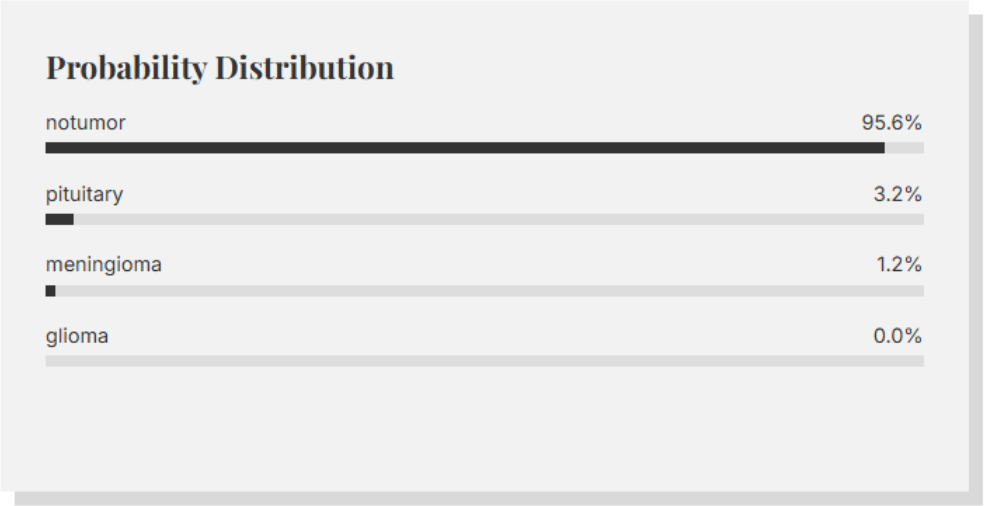



Figure 16: Case probability distribution: *Notumor* strongly dominates over tumor classes.

**Failure analysis:** In this example, the model ignores a visible anomaly and classifies the case as *Notumor*. The main reason is that the image shares the same predominant visual *style* as the healthy class (superior perspective and tight framing), inducing a spurious correlation and generating incorrect confidence in the system (Figures 15 and 16).

# 7 Integration into Web Application/Script

The model was exported in **TensorFlow.js** format and integrated into a web application developed with **HTML, CSS, and JavaScript**, which allows:

- Loading static images.
- Running inference in the browser.
- Showing class probabilities.
- Applying threshold $\theta$ to indicate "uncertainty".

## 7.1 Interface Design

The web application was designed with a clear and modular interface aimed at facilitating result interpretation, prioritizing simple navigation and immediate readability of predictions.

As shown in Figure 17, the *layout* is organized into two main columns. On the left side, interaction elements are grouped: (i) confidence threshold $\theta$ configuration via a slider, (ii) a dataset image library filterable by category, and (iii) an image upload module. On the right side, model results are presented: a classification summary (predicted class and confidence) and a probability distribution as horizontal bars, enabling an intuitive comparison of each class's relative strength. In terms of *look & feel*, a minimalist card-based design was used, with readable typography and whitespace separation to improve visual clarity.

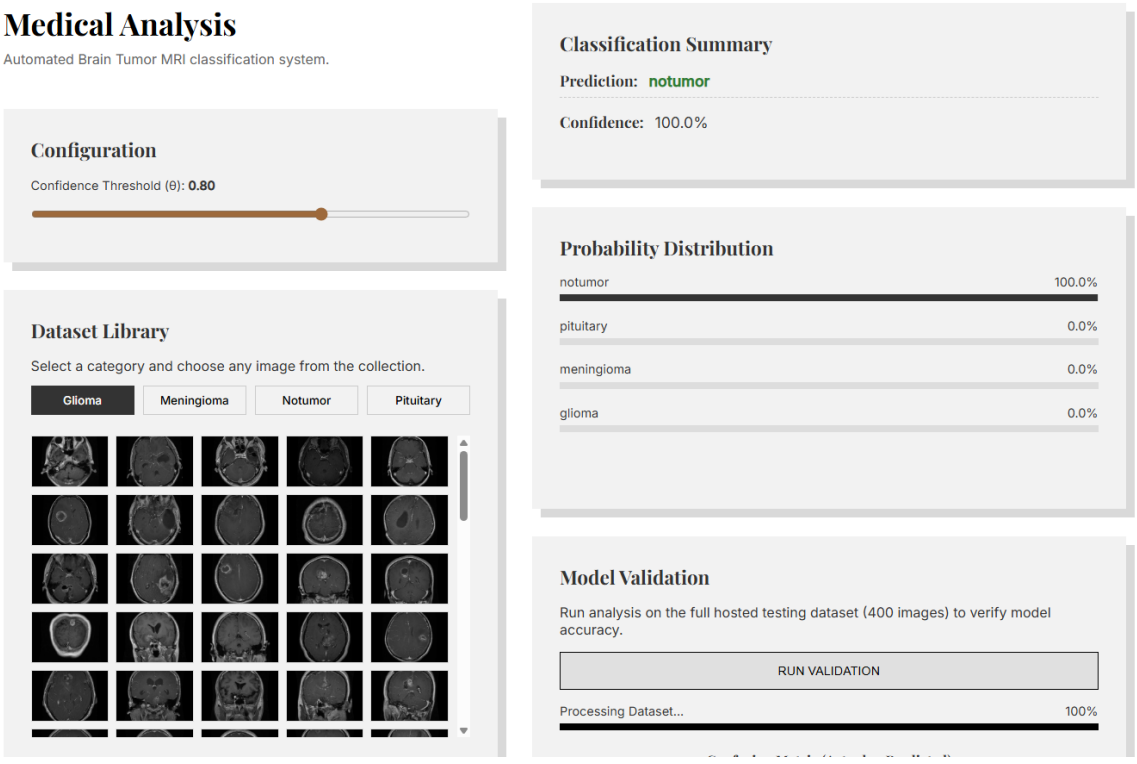
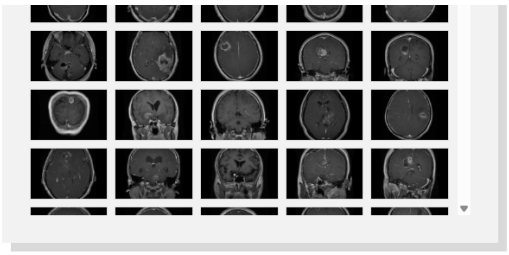



Figure 17: Main view of the web application: threshold $\theta$ configuration, dataset library, and prediction panel with probability distribution.

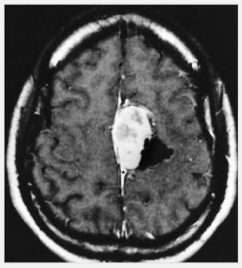
On the other hand, Figure 18 shows the output of the validation module integrated into the application itself. This component allows running the model over the entire test set hosted in a

json file and visualizing aggregated metrics, specifically the confusion matrix and the *classification report*.



**Input Source**

UPLOAD IMAGE



**Model Validation**

Run analysis on the full hosted testing dataset (400 images) to verify model accuracy.

RUN VALIDATION

Processing Dataset... 100%

**Confusion Matrix (Actual vs Predicted)**

| Actual \ Predicted | Glioma | Meningioma | Notumor | Pituitary |
|---|---|---|---|---|
| **Glioma** | **86** | 13 | 0 | 1 |
| **Meningioma** | 6 | **66** | 17 | 11 |
| **Notumor** | 3 | 0 | **96** | 1 |
| **Pituitary** | 0 | 6 | 1 | **93** |

**Classification Report**

| Metric \ Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Glioma** | 0.91 | 0.86 | 0.88 | 100 |
| **Meningioma** | 0.78 | 0.66 | 0.71 | 100 |
| **Notumor** | 0.84 | 0.96 | 0.90 | 100 |
| **Pituitary** | 0.88 | 0.93 | 0.90 | 100 |
| Accuracy | | | **0.85** | 400 |
| **Macro Avg** | 0.85 | 0.85 | 0.85 | 400 |
| **Weighted Avg** | 0.85 | 0.85 | 0.85 | 400 |

Figure 18: Model validation view in the web application and the module for uploading images from your computer: execution on the test dataset and visualization of the confusion matrix and *classification report*.

# 8 Conclusions

After completing the tests, the main conclusion is that an artificial intelligence model is not infallible just because it has high accuracy. Although our main model performs well with 87% accuracy, experiments with different hyperparameters taught us that stability is key. We saw that if the learning rate is too high, the model becomes unstable and stops being reliable, and if we train too little or with batches that are too large, the network fails to learn the details that distinguish one tumor from another.

The most interesting part of the project was discovering that the model can sometimes be misled by things that have nothing to do with medicine. For example, we detected that if a scan is taken from very high above or with a lot of zoom, the model tends to say there is no tumor simply because that image resembles the visual style of healthy images it saw during training. This shows that, in medicine, it is not enough to look at the final accuracy number; it is essential to review the confusion matrix to understand where the system makes mistakes. In the end, the web application we built serves precisely that purpose: to see in real time that human judgment is still necessary to validate what a machine predicts.