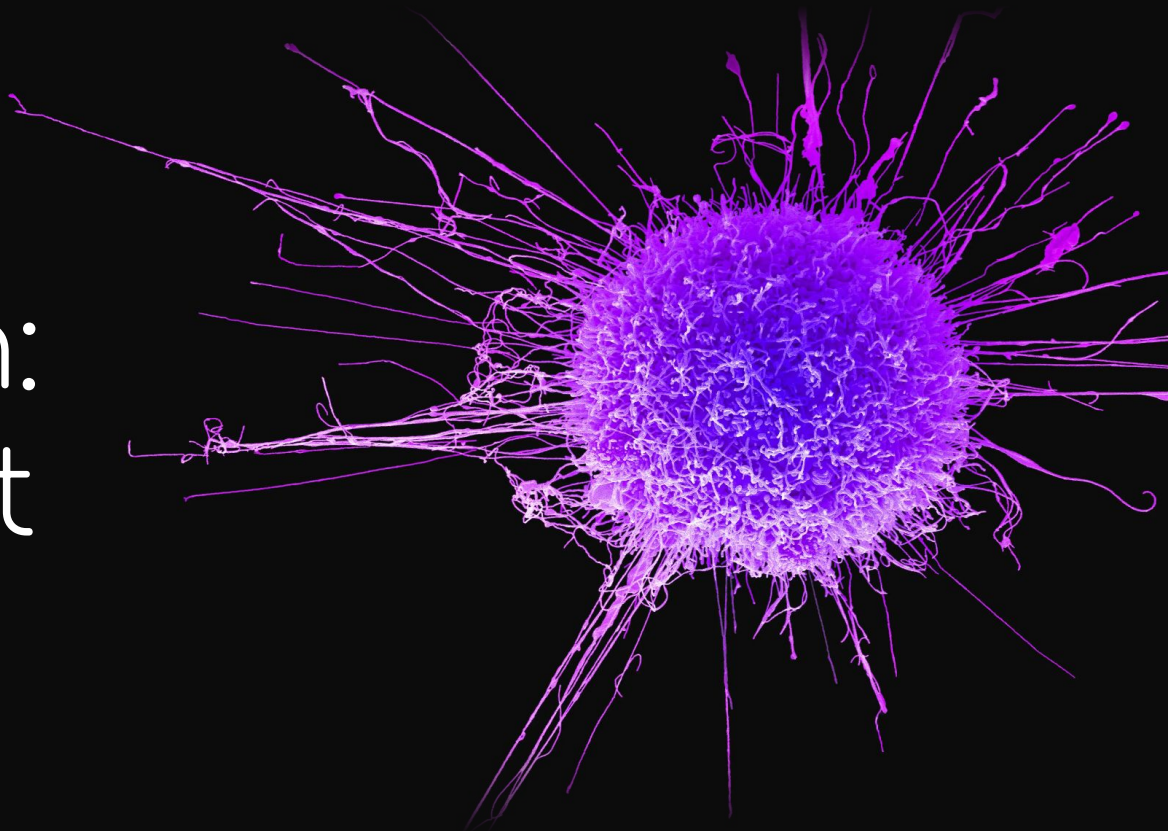


Cancer Prediction: Early Alert

Ramsey Foster
Carlos Vargas
Nicholas Perez
David A. Alvarado Garcia
Hans Jeremy



Project Overview

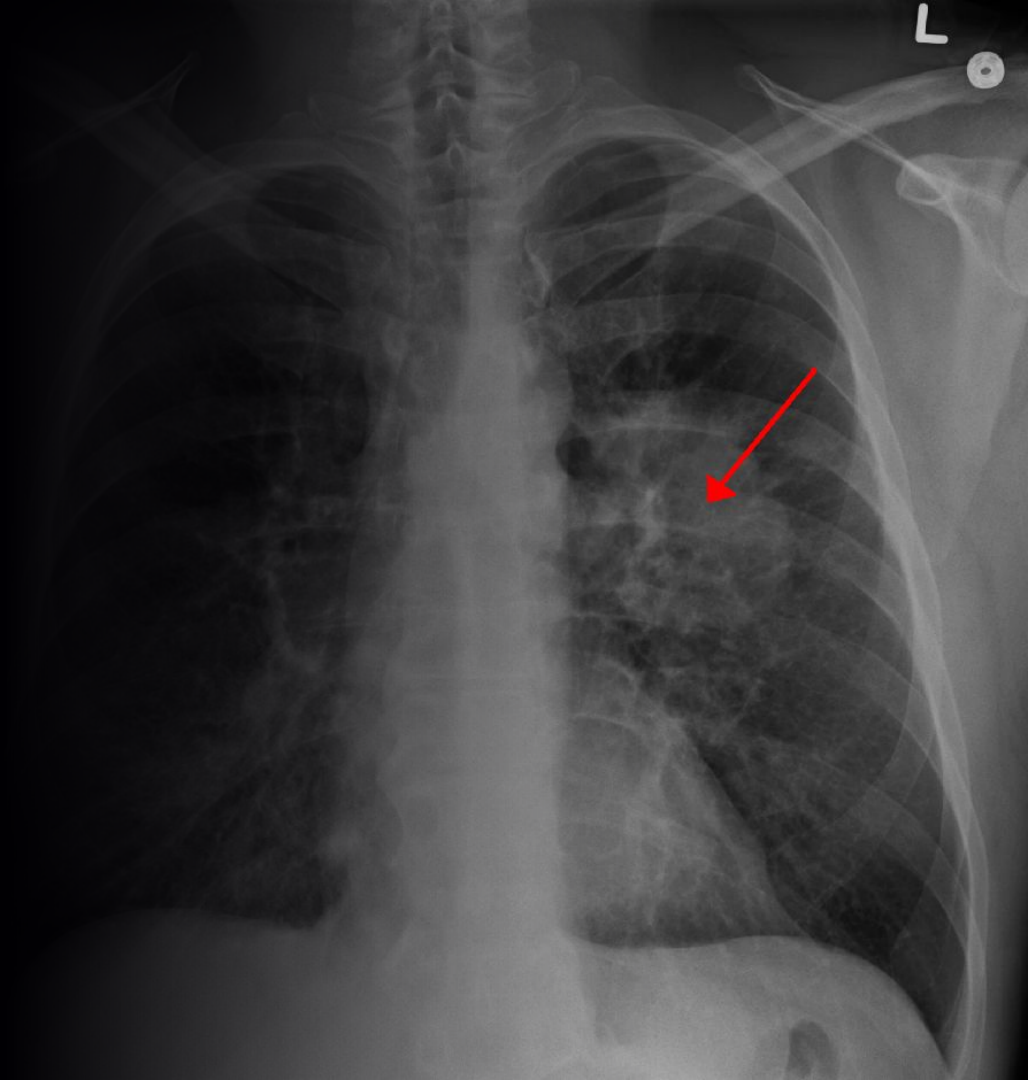
Topic:
Cancer
Prediction

Main Goal:
To create models that can detect
and predict a cancer diagnosis
based on tabular data or x-ray
images

- Objectives:
1. Collect and process datasets
 2. Train and compare multiple algorithms
 3. Evaluate and compare the algorithms

Background

Early detection of cancers can save someone's life as the early stages are crucial for the safe removal of the cancer from the body before it grows any bigger and becomes a problem.



Background

Lab Tests

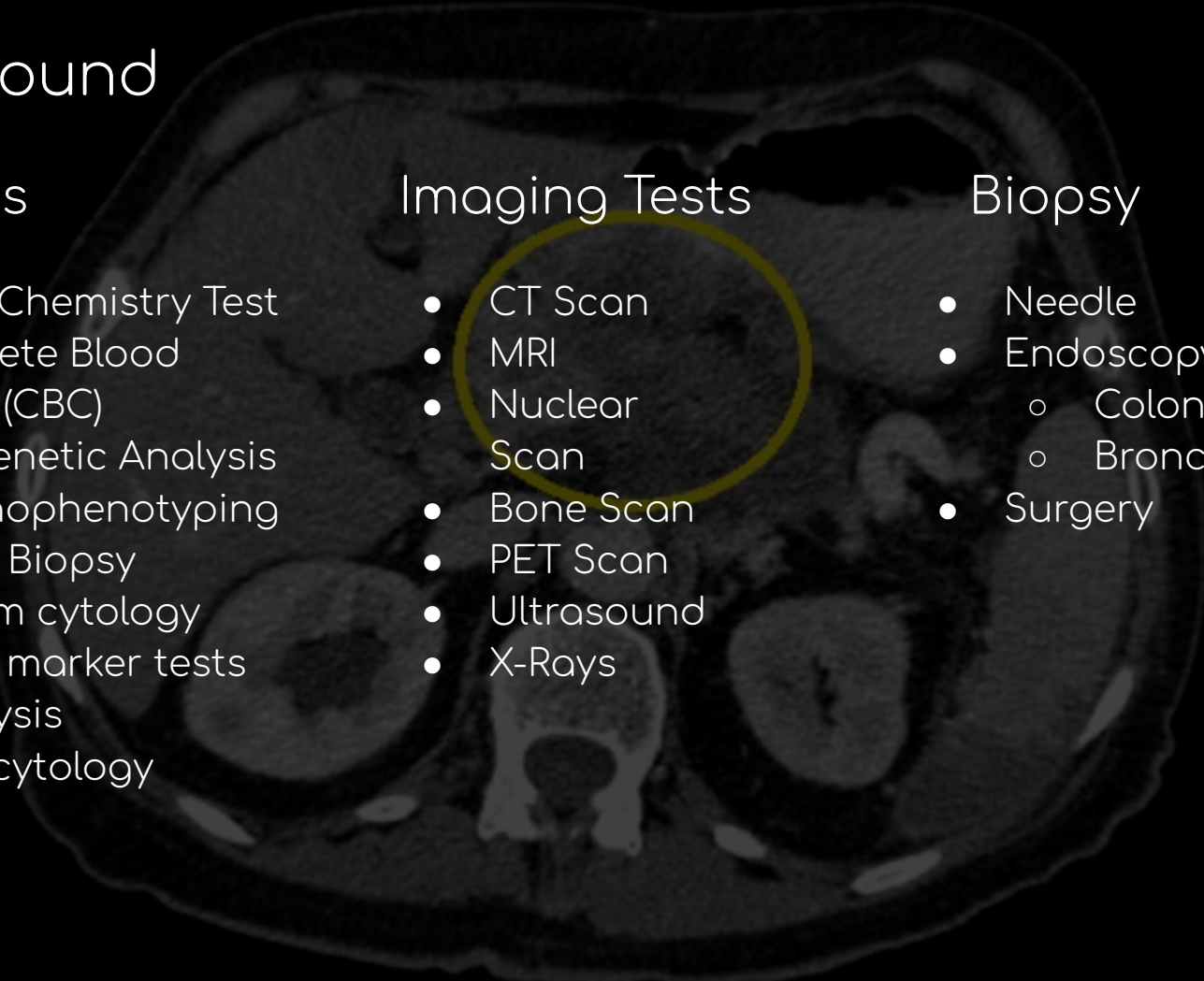
- Blood Chemistry Test
- Complete Blood Count (CBC)
- Cytogenetic Analysis
- Immunophenotyping
- Liquid Biopsy
- Sputum cytology
- Tumor marker tests
- Urinalysis
- Urine cytology

Imaging Tests

- CT Scan
- MRI
- Nuclear Scan
- Bone Scan
- PET Scan
- Ultrasound
- X-Rays

Biopsy

- Needle
 - Colonoscopy
 - Bronchoscopy
- Endoscopy
- Surgery



Datasets Used

Breast Cancer Dataset:

A tabular dataset from Kaggle with a binary diagnosis of benign or malignant

Brain Tumor MRI Dataset:

MRI images from Kaggle with diagnoses of glioma, meningioma, pituitary, and no tumor

Lung Cancer Dataset:

CT images from Kaggle with diagnoses of benign, malignant, or normal

Data Preprocessing

Breast Cancer Dataset:

The input features were scaled using StandardScaler. This standardized each feature for a comparable scale and to train the model more effectively

Data Preprocessing

Lung/Brain Cancer Datasets:

Both the lung CT and brain MRI images were converted to a consistent PNG format, resize pixels, and normalized to ensure uniform brightness and scale. Light augmentations like rotations and flips were applied to help the model generalize across different imaging angles and variations in the data.

Breast Cancer Dataset

Methodology

Model Training:

- Implemented:
 - Multilayer Perceptron
 - Support Vector Classifier
 - K-Nearest Neighbors
 - Logistic Regression
 - Gaussian naive Bayes

Evaluation and Visualization:

- Visualizations:
 - Confusion Matrices
 - ROC curves
- Evaluation Scores:
 - Accuracy
 - Precision
 - Recall
 - F1 Score
 - Kappa

Breast Cancer Dataset

Methodology

Data:

- Input features:
 - Everything except ID and diagnoses
 - Mean of Radius, Standard Error of Texture, etc.
- Output features:
 - Diagnoses
- 80/20 split between training and testing

Breast Cancer Dataset

Implementation

Libraries:

- Imported:
 - Numpy
 - Pandas
 - Kagglehub
 - Matplotlib
 - Sci-Kit Learn
 - Metrics
 - Different Algorithms

Breast Cancer Dataset

Implementation

Code Snippets:

```
# hidden layers almost match the amount of input features (32)
# Relu activation handles continuous features and avoids vanishing gradients
bcMLPModel = MLPClassifier(hidden_layer_sizes=(20, 10), activation='relu', max_iter=500)
# 'l1' is because number of input features is high which performs feature
# selection by zeroing out less useful coefficients
bcSVCModel = LinearSVC(C=0.1, max_iter=500, penalty='l1')
# k=5 avoids overfitting
bcKNNModel = KNeighborsClassifier(n_neighbors=5)
# 'l2' prevents overfitting and controls weights
bcLRModel = LogisticRegression(penalty='l2')
bcNBModel = GaussianNB()
```

Breast Cancer Dataset

Output Screenshots

```
MLP Diagnoses predictions:      ['M' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B']
SVM Diagnoses predictions:      ['M' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B']
KNN Diagnoses predictions:      ['M' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B']
Logistic Regression Diagnoses predictions:  ['M' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B']
Gaussian Bayes Diagnoses predictions:      ['M' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B']
Actual diagnoses:               ['M' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'B']
```

```
MLP accuracy:  0.97368
SVM accuracy:  0.98246
KNN accuracy:  0.96491
Logistic Regression accuracy:  0.98246
Gaussian Bayes accuracy:  0.91228
```

```
MLP Model precision:  0.94595
SVM Model precision:  0.97222
KNN Model precision:  0.94444
Logistic Regression Model precision:  0.97222
Gaussian Bayes Model precision:  0.825
```

```
MLP Model recall:  0.97222
SVM Model recall:  0.97222
KNN Model recall:  0.94444
Logistic Regression Model recall:  0.97222
Gaussian Bayes Model recall:  0.91667
```

```
MLP Model F1 score:  0.9589
SVM Model F1 score:  0.97222
KNN Model F1 score:  0.94444
Logistic Regression Model F1 score:  0.97222
Gaussian Bayes Model F1 score:  0.86842
```

```
MLP Model kappa:  0.93955
SVM Model kappa:  0.9594
KNN Model kappa:  0.9188
Logistic Regression Model kappa:  0.9594
Gaussian Bayes Model kappa:  0.8029
```

Breast Cancer Dataset

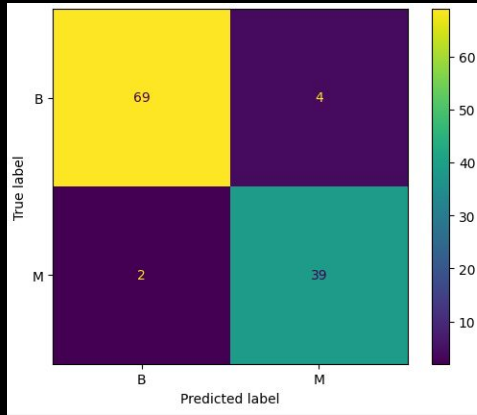
Evaluation Metrics

	Multilayer Perceptron	Support Vector Classifier	K-Nearest Neighbors	Logistic Regression	Gaussian naive Bayes
Accuracy	94.737%	96.491%	95.614%	96.491%	90.351%
Precision	90.698%	95.122%	100.000%	95.122%	89.474%
Recall	95.122%	95.122%	87.805%	95.122%	82.927%
F ₁ Score	92.857%	95.122%	93.506%	95.122%	86.076%
Kappa	88.694%	92.382%	90.216%	92.382%	78.710%

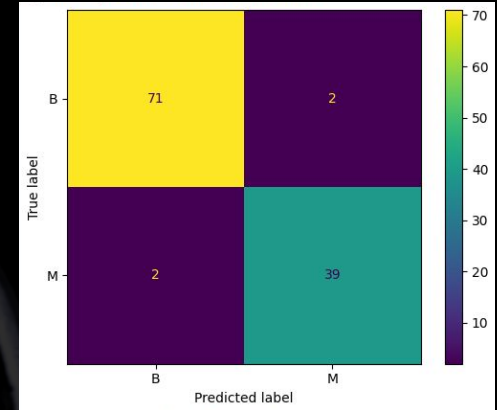
Breast Cancer Dataset

Confusion Matrices

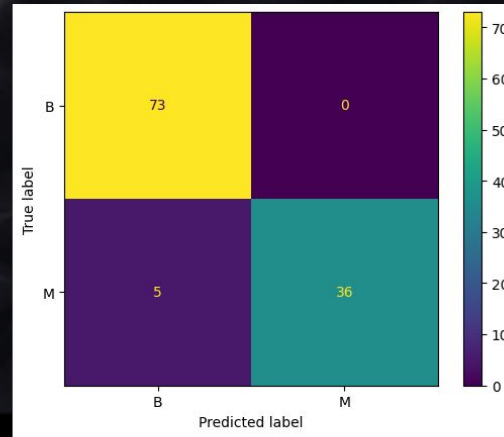
Multilayer Perceptron



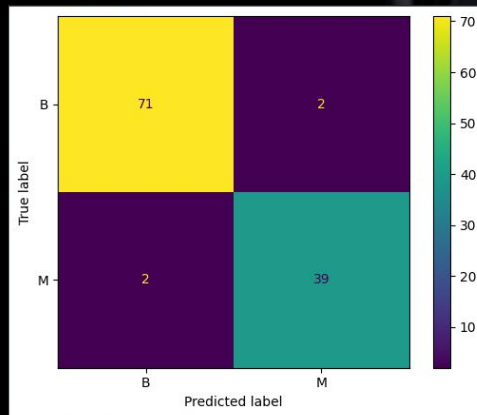
Logistic Regression



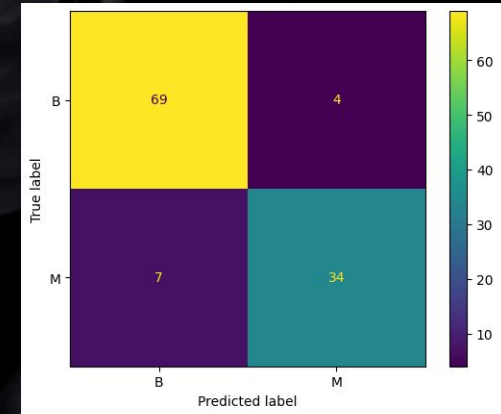
K-Nearest Neighbors



Support Vector Classifier

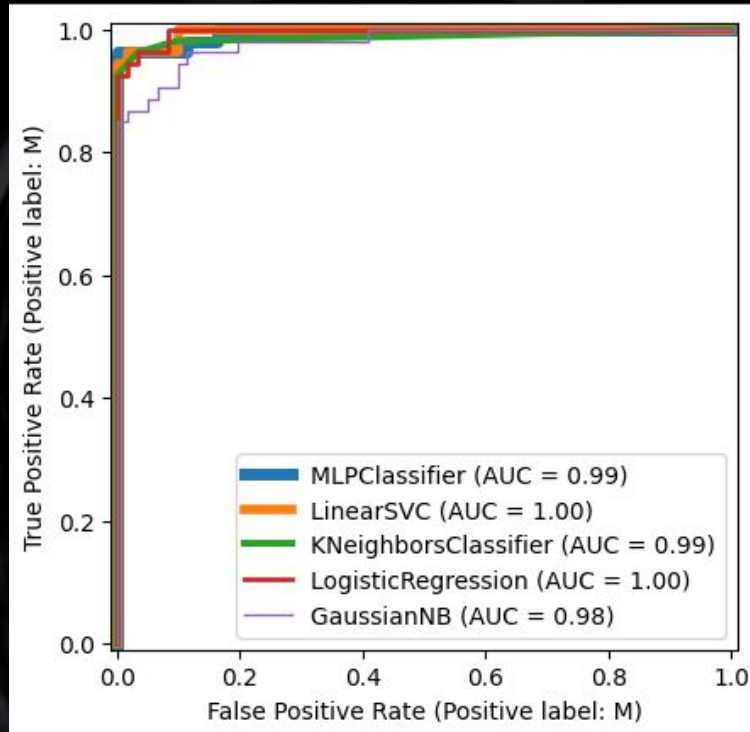


Gaussian naive Bayes



Breast Cancer Dataset

ROC Curves



Lung/Brain Cancer Datasets

Methodology

Model Training:

- Implemented:
 - ResNet50 CNN
 - HOG (Histogram of Oriented Gradients) + SVM (Support Vector Machine)

Evaluation and Visualization:

- Visualizations:
 - Confusion Matrices
- Evaluation Scores:
 - Accuracy
 - Precision
 - Recall
 - F1 Score
 - Kappa

Lung/Brain Cancer Datasets

Methodology

Data:

- Input features:
 - ResNet50 CNN
 - Full RGB Images
 - HOG+SVM
 - 1D Vector with ~1000 - ~4000 features per images (NOT the image itself)
- Output features:
 - CNN & HOG+SVM
 - Predicted diagnosis on image uploaded
- 80/20 split between training and testing

Lung/Brain Cancer Datasets

Implementation

Libraries:

- Imported:
 - ResNet50 CNN
 - TensorFlow/Keras
 - NumPy
 - Matplotlib
 - SciKit Learn
 - HOG+SVM
 - OpenCV
 - SciKit Image
 - SciKit Learn
 - NumPy

Lung/Brain Cancer Datasets

Implementation

Code Snippets(ResNet50 CNN (Lung on Left)(Brain on Right)):

```
from tensorflow.keras.applications import ResNet50
from tensorflow.keras import layers, Model

base_resnet = ResNet50(
    include_top=False,
    weights="imagenet",
    input_shape=(224, 224, 3)
)

base_resnet.trainable = False # Stage 1

inputs = layers.Input(shape=(224, 224, 3))
x = augmentation(inputs)
x = base_resnet(x, training=False)
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dropout(0.3)(x)
outputs = layers.Dense(len(lung_classes), activation="softmax")(x)

resnet_model = Model(inputs, outputs)

resnet_model.compile(
    optimizer=tf.keras.optimizers.Adam(1e-3),
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)

resnet_model.summary()
```

```
from tensorflow.keras.applications import ResNet50
from tensorflow.keras import layers, Model

base_resnet_brain = ResNet50(
    include_top=False,
    weights="imagenet",
    input_shape=(224, 224, 3)
)

base_resnet_brain.trainable = False # Stage 1

inputs = layers.Input(shape=(224, 224, 3))
x = brain_augmentation(inputs)
x = base_resnet_brain(x, training=False)
x = layers.GlobalAveragePooling2D()(x)
x = layers.BatchNormalization()(x)
x = layers.Dropout(0.3)(x)

outputs = layers.Dense(len(brain_classes), activation="softmax")(x)

brain_model = Model(inputs, outputs)

brain_model.compile(
    optimizer=tf.keras.optimizers.Adam(1e-3),
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)

brain_model.summary()
```

Lung/Brain Cancer Datasets

Implementation

Code Snippets (HOG+SVM (Both Lung and Brain):

```
import cv2
import numpy as np
from skimage.feature import hog

X = []
y = []

def extract_hog_features(img_path):
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
    img = cv2.resize(img, (128, 128))

    features, _ = hog(
        img,
        orientations=9,
        pixels_per_cell=(8, 8),
        cells_per_block=(2, 2),
        block_norm='L2-Hys',
        visualize=True
    )

    return features

print("Extracting HOG features")

for label_index, cls in enumerate(CLASS_NAMES):
    folder = os.path.join(DATA_PATH, cls)
    for file in os.listdir(folder):
        if file.lower().endswith(".png"):
            path = os.path.join(folder, file)
```

```
        features = extract_hog_features(path)
        X.append(features)
        y.append(label_index)
```

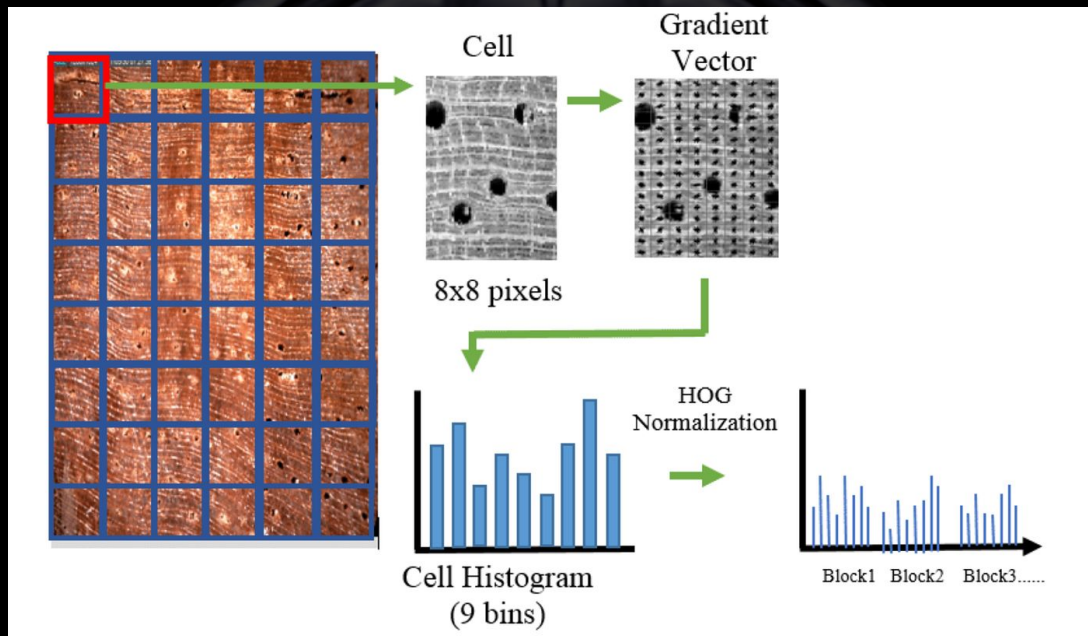
```
X = np.array(X)
y = np.array(y)
```

```
print("Feature extraction complete")
print("Feature vectors:", X.shape)
print("Labels:", y.shape)
```

Lung/Brain Cancer Datasets

Implementation

Photo Snippet (HOG+SVM (Both Lung and Brain):



Lung/Brain Cancer Datasets

Evaluation Metrics

Lung Dataset

	ResNet50 CNN	HOG+SVM
Accuracy	97.539%	99.091%
Precision	94.004%	99.306%
Recall	97.94%	98.21%
F ₁ Score	95.658%	98.742%
Kappa	95.825%	98.433%

Brain Dataset

	ResNet50 CNN	HOG+SVM
Accuracy	76.138%	93.876%
Precision	78.911%	93.659%
Recall	74.577%	93.652%
F ₁ Score	73.42%	93.639%
Kappa	67.885%	98.13%

Lung/Brain Cancer Datasets

Evaluation Metrics

Lung Dataset Samples (ResNet50 CNN):

Predictions: [1 1 0 1 2 1 2 2 1 2 2 1 0 0 2]

True Labels: [1 1 0 1 2 1 2 2 2 2 2 1 2 0 2]

Brain Dataset Samples (ResNet50 CNN):

Predictions: [2 0 2 2 0 2 3 3 1 3 3 3 3 2 2]

True Labels: [2 0 2 2 0 2 3 3 1 1 3 3 0 2 2]

Lung Dataset Samples (HOG+SVM):

Predictions: [2 2 1 2 1 2 1 2 2 1 1 1 0 1 1]

True Labels: [2 2 1 2 1 2 1 2 2 1 1 1 0 1 1]

Brain Dataset Samples (HOG+SVM):

Predictions: [1 3 1 3 2 3 0 0 3 2 1 1 2 0 1]

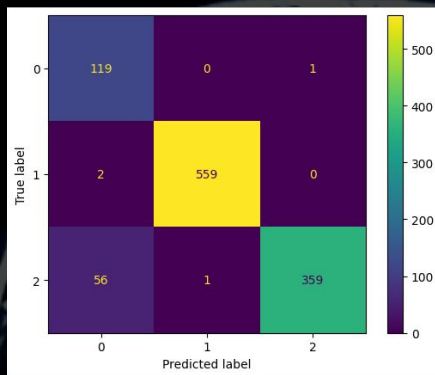
True Labels: [1 3 1 3 2 3 0 0 3 2 1 1 2 0 1]

Lung/Brain Cancer Datasets

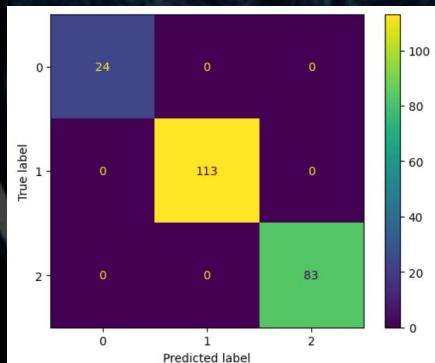
Confusion Matrices

Lung Dataset:

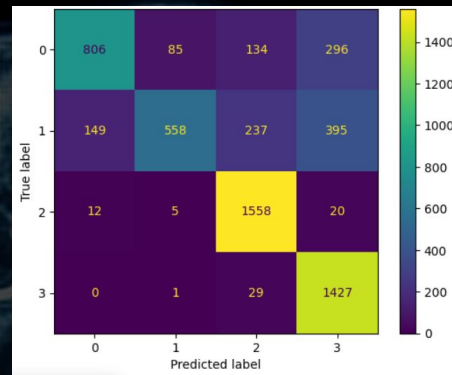
0 = Benign
1 = Malignant
2 = Normal



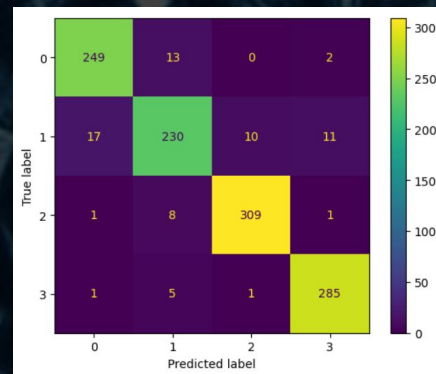
ResNet50 CNN Lung



HOG+SVM Lung



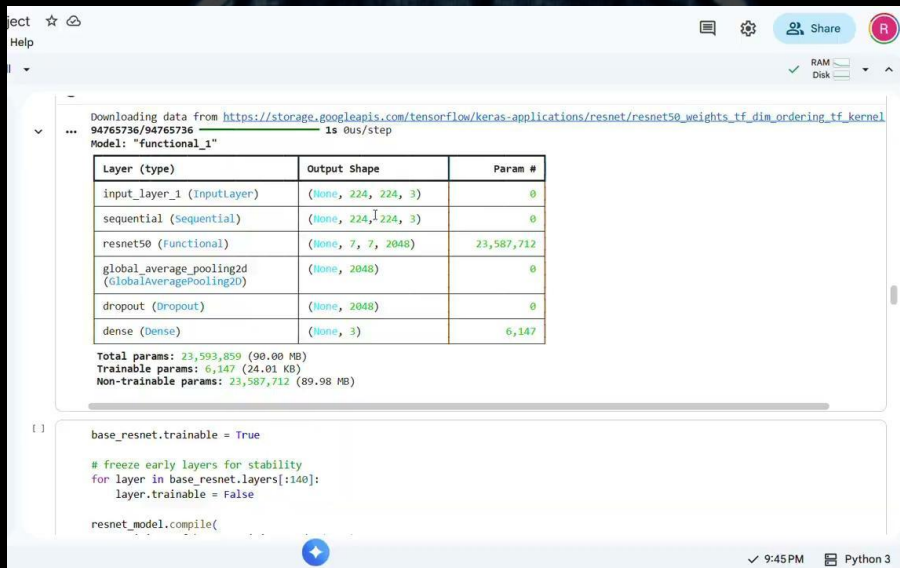
ResNet50 Brain



HOG+SVM Brain

Lung/Brain Cancer Datasets

Demo (ResNet50 CNN):



Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels/94765736/94765736 1s 0us/step
Model: "functional_1"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
sequential (Sequential)	(None, 224, 224, 3)	0
resnet50 (Functional)	(None, 7, 7, 2048)	23,587,712
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dropout (Dropout)	(None, 2048)	0
dense (Dense)	(None, 3)	6,147

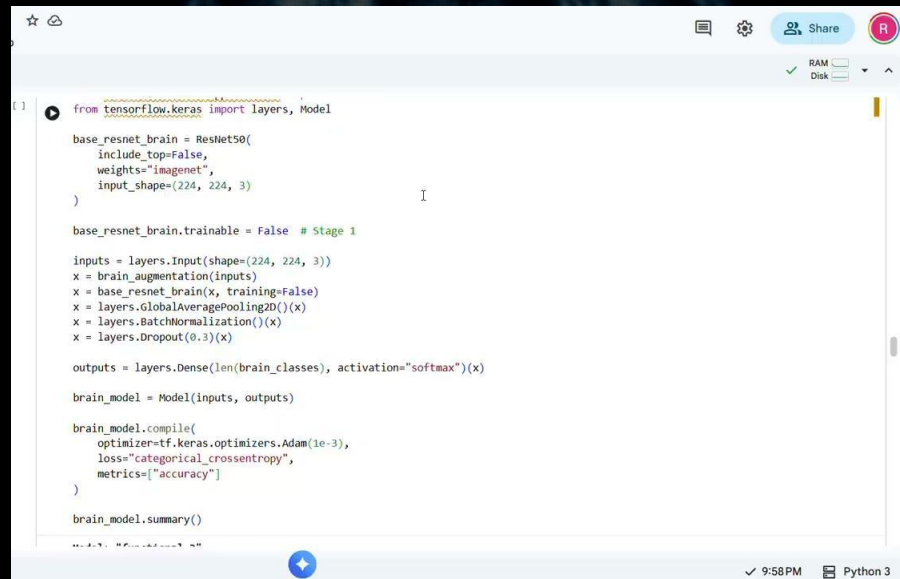
Total params: 23,593,859 (90.00 MB)
Trainable params: 6,147 (24.01 KB)
Non-trainable params: 23,587,712 (89.98 MB)

```
base_resnet.trainable = True

# freeze early layers for stability
for layer in base_resnet.layers[:140]:
    layer.trainable = False

resnet_model.compile(
```

ResNet50 CNN Lung



```
from tensorflow.keras import layers, Model

base_resnet_brain = ResNet50(
    include_top=False,
    weights='imagenet',
    input_shape=(224, 224, 3)
)

base_resnet_brain.trainable = False # Stage 1

inputs = layers.Input(shape=(224, 224, 3))
x = brain_augmentation(inputs)
x = base_resnet_brain(x, training=False)
x = layers.GlobalAveragePooling2D()(x)
x = layers.BatchNormalization()(x)
x = layers.Dropout(0.3)(x)

outputs = layers.Dense(len(brain_classes), activation='softmax')(x)

brain_model = Model(inputs, outputs)

brain_model.compile(
    optimizer=tf.keras.optimizers.Adam(1e-3),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

brain_model.summary()
```

ResNet50 Brain

Lung/Brain Cancer Datasets

Demo (HOG+SVM):

```
[30] In
def extract_hog_features(img_path):
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
    img = cv2.resize(img, (128, 128))

    features, _ = hog(
        img,
        orientations=9,
        pixels_per_cell=(8, 8),
        cells_per_block=(2, 2),
        block_norm='L2-Hys',
        visualize=True
    )

    return features

print("Extracting HOG features")

for label_index, cls in enumerate(CLASS_NAMES):
    folder = os.path.join(DATA_PATH, cls)
    for file in os.listdir(folder):
        if file.lower().endswith(".png"):
            path = os.path.join(folder, file)
            features = extract_hog_features(path)
            X.append(features)
            y.append(label_index)

X = np.array(X)
y = np.array(y)
```

✓ 10:26 PM Python 3

HOG+SVM Lung

```
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (1.5.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (3.6.0)

import os

# Choose dataset: "lung" or "brain"
DATASET = "brain"

if DATASET == "lung":
    DATA_PATH = "/content/lung_training_clean" # your cleaned lung dataset
    CLASS_NAMES = ["benign_lung", "malignant_lung", "normal_lung"]

elif DATASET == "brain":
    DATA_PATH = "/content/Image Dataset/Training_Data"
    CLASS_NAMES = ["glioma", "meningioma", "notumor", "pituitary"]

print("Using dataset:", DATA_PATH)
print("Classes:", CLASS_NAMES)

... Using dataset: /content/Image Dataset/Training_Data
Classes: ['glioma', 'meningioma', 'notumor', 'pituitary']

import cv2
import numpy as np
from skimage.feature import hog

X = []
y = []
```

✓ 10:22 PM Python 3

HOG+SVM Brain

Discussion & Conclusion

Data:

- ResNet50 CNN:
 - Did fairly poorer with lung and brain datasets compared to HOG+SVM
 - The brain dataset did significantly poor due to the numerous angles in the MRI Scans between all classes
- HOG+SVM:
 - Given our results, both lung and brain datasets worked best with HOG+SVM
 - We believe this to be the case due to HOG extracting the features rather than finding them the way CNN does
- Logistic Regression:
 - Performed well with breast cancer data as it provided accurate prediction based off tabular data similar to how the HOG+SVM performed

Overall SVM+HOG worked best for our image datasets and Logistical Regression did the best with tabular data. Going forward we would continue to make our model using the HOG+SVM as we believe it to combine all our datasets together.

Limitations and Future Works

- Limitations:
 - Limited and Imbalanced Dataset Sizes
 - HOG + SVM Cannot Capture Deep Medical Features
 - ResNet50 Requires Significant Compute Power
 - 2D Images Instead of 3D Volumes
- Future Works:
 - Combine the Models into One Cohesive Area
 - Add Square Indicators to Result Images
 - Add More Cancer Type Variations to the Model

References

1. <https://www.cancer.gov/about-cancer/diagnosis-staging/diagnosis>
2. <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>
3. <https://www.kaggle.com/datasets/adityamahimkar/iqothnccd-lung-cancer-dataset>
4. <https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset>
5. <https://www.cancer.gov/about-cancer/diagnosis-staging/diagnosis>
6. <https://pmc.ncbi.nlm.nih.gov/articles/PMC10546033/>
7. <https://www.sciencedirect.com/topics/computer-science/resnet50>
8. <https://pmc.ncbi.nlm.nih.gov/articles/PMC4220242/>
9. <https://www.grqpc.com/gray-ritter-graham-blog/diagnostic-errors-involving-mri-and-ct-scans>
10. <https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/ct-vs-mri-vs-xray>