PROGRESS PREDICTION
4° WEEK

Code Review

Extract Features

Generate csv file with informaton generated from images

1. Metadata extracted SliceThickness and PixelSpacing

2. Make Lung mask
   a. Normalize image → remove mean and divide by std
   b. Renormalize washed images → sub light/dark pixels with mean
   c. K-means to separate foreground and background
   d. Erosion → eliminate noise/small details with a 3x3 filter
   e. Dilation → reconstruct principal areas through a 8x8 filter
   ....

Code Review

Extract Features

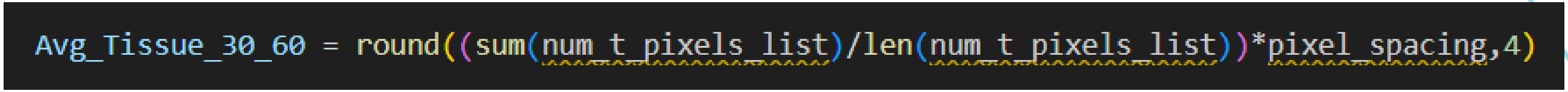
....

a. Label creation (skimage) → assign labels for each portion
b. Compute geometrical attributes (area,bounding box)
c. Select good bounding boxes → eliminate too big/small areas
d. Fill lung masks → 1 for lungs, 0 elsewhere
e. Compute lung area
f. Calculate tissue mask and extract features (lung without border)

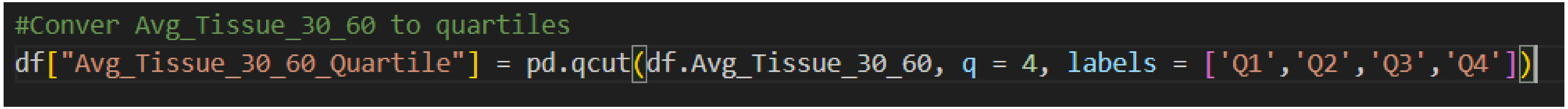
3. Join extracted features to metadata and known data

Code Review

Quantile definition



```
Avg_Tissue_30_60 = round((sum(num_t_pixels_list)/len(num_t_pixels_list))*pixel_spacing,4)
```



```
#Conver Avg_Tissue_30_60 to quartiles
df["Avg_Tissue_30_60_Quartile"] = pd.qcut(df.Avg_Tissue_30_60, q = 4, labels = ['Q1','Q2','Q3','Q4'])
```

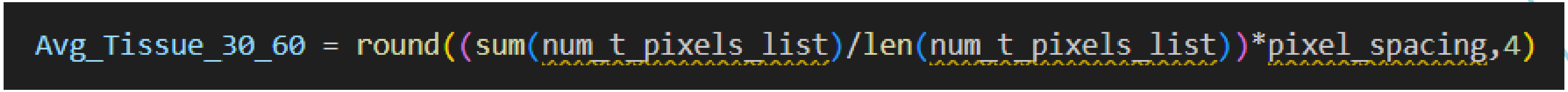
Uses Avg_tissue_30_60 to define quantile groups and define categorical values.
Computed through:

- **num_t_pixels_list : list of the number of tissue pixels detected in image slices between 30% and 60% of the lung height**
- **pixel_spacing : metadata**

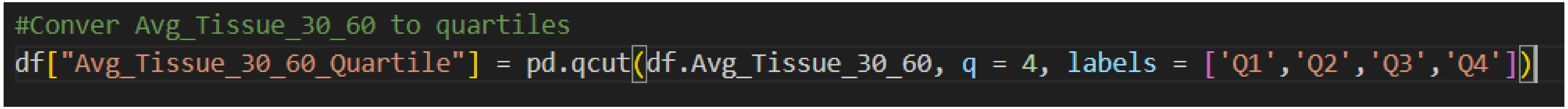
So it’s the average tissue area (in mm^2).

Code Review

Quantile definition



```
 Avg_Tissue_30_60 = round((sum(num_t_pixels_list)/len(num_t_pixels_list))*pixel_spacing,4)
```



```
#Conver Avg_Tissue_30_60 to quartiles
df["Avg_Tissue_30_60_Quartile"] = pd.qcut(df.Avg_Tissue_30_60, q = 4, labels = ['Q1','Q2','Q3','Q4'])
```

“pd.qcut” used to divide data into 4 groups with the same amount of data 4 groups based on percentiles (25,50,75).

labels=['Q1','Q2','Q3','Q4'] assigns quartile names:

- **Q1: lowest 25% of average tissue areas**
- **Q2: 25–50%**
- **Q3: 50–75%**
- **Q4: top 25% (largest average tissue areas)**

Code Review

Modeling 1

For each patient p in the train set:

- Fits a linear regression and saves the slope (a), tab values and patient

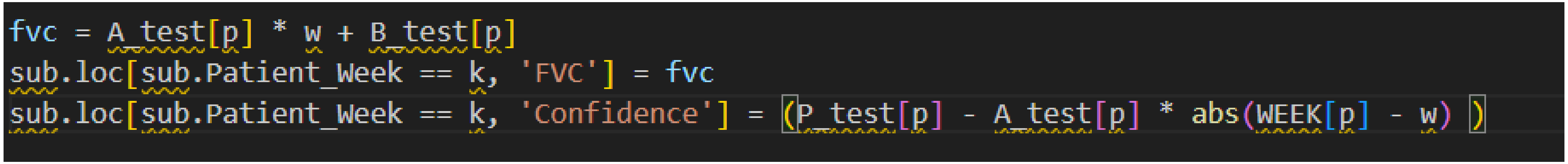
Generates 5 folds and split patients between these 5 folds.
For each iteration chooses 4 for training and 1 for validation.

Per iteration it builds a new efficient model (so for each iteration it trains the model on a slightly different training set).
Per iteration each patient in the test set, gets slices and tabular values and predict a slope for each slice , choosing the slope through the quantile selected for that fold.

Code Review

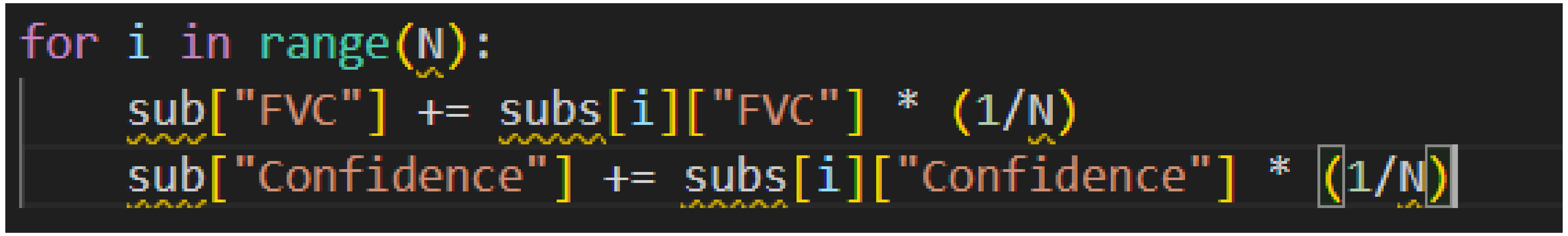
Modeling 1

Having the predicted slope, we can predict the FVC and Confidence for the week defined in the sample_submission csv.
How:



```
fvc = A_test[p] * w + B_test[p]
sub.loc[sub.Patient_Week == k, 'FVC'] = fvc
sub.loc[sub.Patient_Week == k, 'Confidence'] = (P_test[p] - A_test[p] * abs(WEEK[p] - w) )
```

In the end we will have a different prediction for each iteration and an average will be made.



```
for i in range(N):
    sub["FVC"] += subs[i]["FVC"] * (1/N)
    sub["Confidence"] += subs[i]["Confidence"] * (1/N)
```

Code Review

Preparation data

Prepare data:

- Add a train/test/val column
- Add minimum week column (earliest visit for patient)
- Baseline FVC column
- Baseline Percent column
- Add column to indicate time passed from baseline visit
- One-hot encoder for Sex and SmokingStatus
- Add image features extracted from image

Merge all data, handle outliers and noise and normalize.

Code Review

Modeling 2

The models final output is formed by three values:

[y_lower, y_pred, y_upper]

Representing the lower quantile,median and upper quantile estimates of FVC for a patient at a given week.

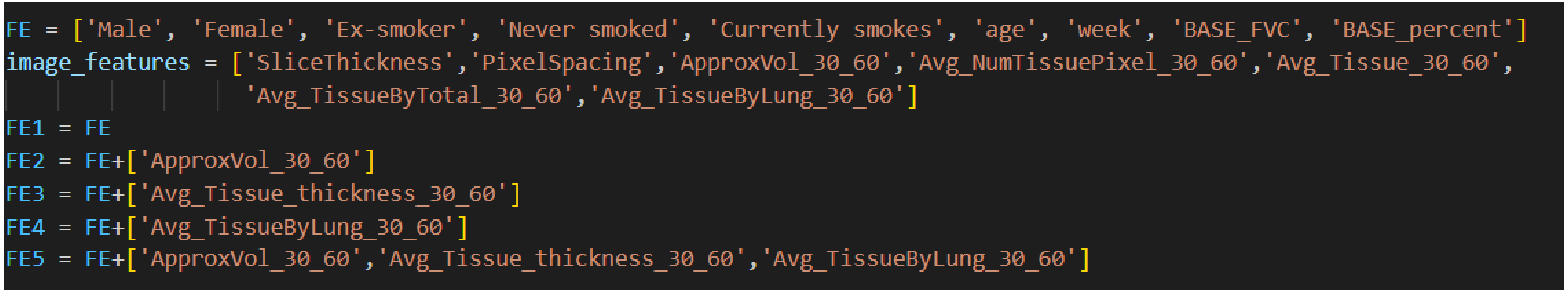
The model uses a combined loss (mloss):

- qloss → encourages predictions for each quantile to bracket the true value correctly
- score → approximates the laplace log-likelihood

Code Review

Modeling 2

5 Neural Networks, each work on a slightly different feature set:



```
FE = ['Male', 'Female', 'Ex-smoker', 'Never smoked', 'Currently smokes', 'age', 'week', 'BASE_FVC', 'BASE_percent']
image_features = ['SliceThickness','PixelSpacing','ApproxVol_30_60','Avg_NumTissuePixel_30_60','Avg_Tissue_30_60',
                  'Avg_TissueByTotal_30_60','Avg_TissueByLung_30_60']
FE1 = FE
FE2 = FE+['ApproxVol_30_60']
FE3 = FE+['Avg_Tissue_thickness_30_60']
FE4 = FE+['Avg_TissueByLung_30_60']
FE5 = FE+['ApproxVol_30_60','Avg_Tissue_thickness_30_60','Avg_TissueByLung_30_60']
```

Collects all predictions and search for optimal ensemble weights - in a brute-force way - across the 5 models.

Code Review

Modeling 2

The final ouput FVC and Confidence is given by:

- FVC : median value (y_true)

- Confidence: y_upper - y_lower

Then finally it blends the predictions to the first model:

- 40% image-based model
- 60% metadata model

Comparison to other approaches

- 5th Place:

Small network with only tabular data
Inputs → [WeekInit, WeekTarget, WeekDiff, FVC, Percent, Age, Sex, CurrentlySmokes, Ex-smoker, Never Smoked]

- 6th Place:

Each measurement in the dataset is treated as if it were a baseline measurement. A new feature week_passed is created and extracted image features as base data. Used 5 models and weighted them
[Lasso, Ridge, ElasticNet, SVM, NN] = [0.68573749, 0., 0., 0.07551167, 0.23750526]

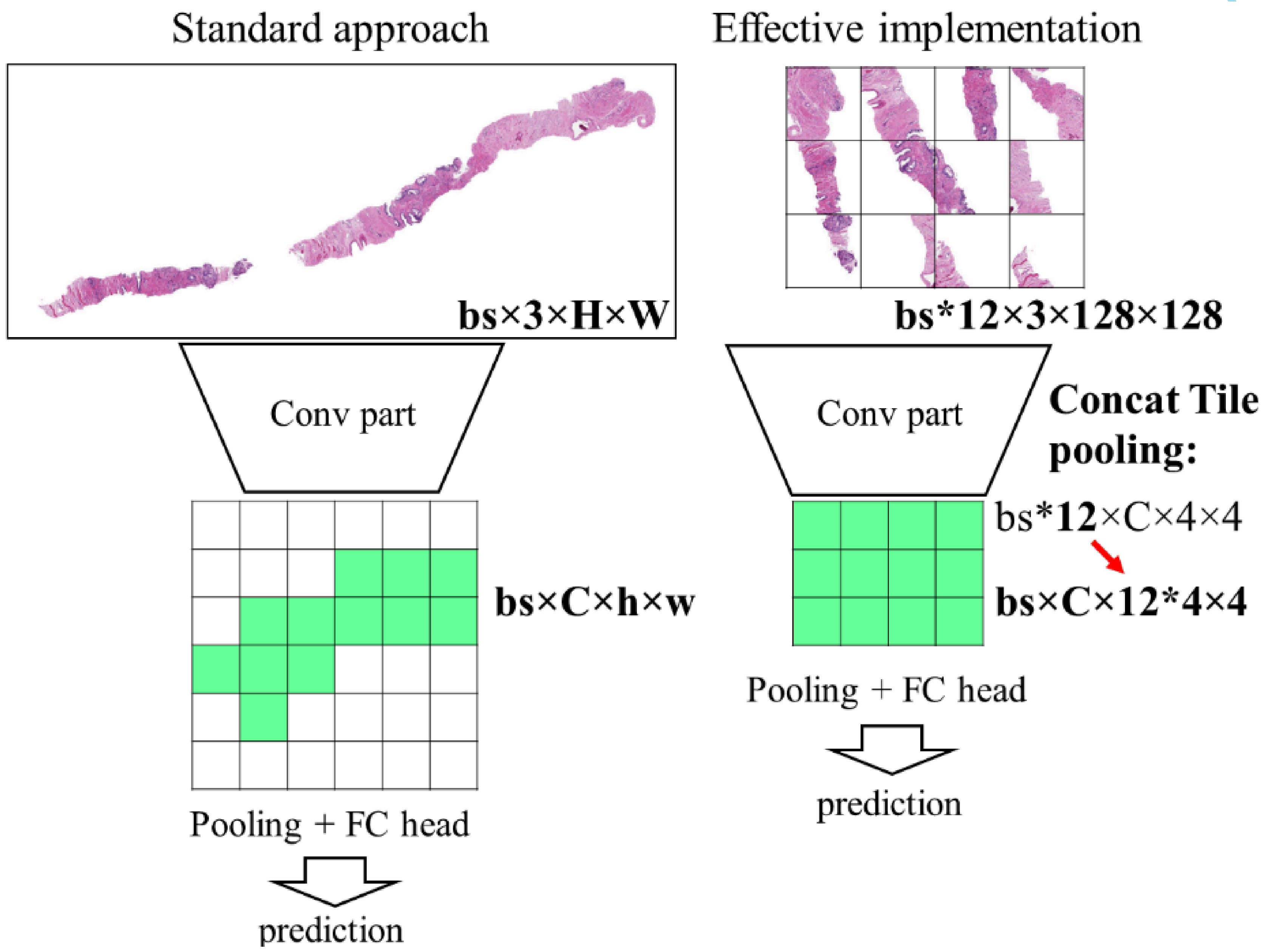
- 9th Place:

Use of Concatenate Tile Pooling approach for 2D CT scans, aggregates information across multiple CT layers and assigns a single label to the entire scan.

Concatenate Tile Pooling

Instead of passing an entire image as an input, N tiles are selected from each image based on the number of tissue pixels and passed independently through the convolutional part.

The outputs of the convolutional part is concatenated in a large single map for each image preceding pooling and FC head .


Standard approach
Effective implementation
bs×3×H×W
bs*12×3×128×128
Conv part
Conv part
Concat Tile pooling:
bs×C×h×w
bs*12×C×4×4
bs×C×12*4×4
Pooling + FC head
Pooling + FC head
prediction
prediction