**MRI-PET Image Fusion Using CNN**

1. **Introduction**

Image fusion is the process of merging information from many different images into a single picture. This creates a more detailed and usable image, which is frequently used in artificial intelligence, satellite imagery, and imaging for medical purposes. Depending on the application, numerous techniques can be utilized, such as average values of pixels or converting pictures into different fields.

Image fusion involves the combination of data from various pictures in order to offer an additional comprehensive and informed output. The technique is separated into two stages, which are pre-fusion and post-fusion.

* 1. **Pre-Fusion**

Pre-fusion actions include preparing the input data pictures to facilitate successful merging. This includes activities like picture preparation to improve quality, rectify distortions, and minimize noise. Furthermore, extraction of features may occur, dividing relevant information from every image that will help in the fusion process. Pre-fusion sets the foundation for enhanced information synthesis by refining and supplementing the input images ahead of time.

If we give a example of pre-fusion that image enhancement. Image enhancement is prior to fusion, pictures may be improved in quality. Consider a particular scenario in global imagery images from numerous cameras have value resolutions. During the pre-fusion stage, the lower-resolution imagine can be upscaled or enhanced in order to complement the higher-resolution picture. This assures that both input pictures have similar clarity, resulting in a more successful fusion operation.

* 1. **Post-Fusion**

On the opposite stake, post-fusion steps happen after the process of fusion is complete. This step concentrates on refining and obtaining relevant information from the merged images. For improving the entire output, techniques like further improvement, sharpening, and contrast alterations may be used. Furthermore, specialized analysis or extraction of features may be performed on the fused picture to extract essential information for a specific application. In essence, post-fusion seeks to optimize the usability of the fused picture by ensuring that it fulfills the precise aims and needs of the job. By digging into the pre-fusion and post-fusion complexity, investigators and practitioners may precisely adjust each part of the picture fusion process, leading to more precise and pertinent outcomes.

Feature Extraction from a Fused Picture is a method that uses post-fusion example. This method is utilized after integrating numerous medical pictures, such as MRI and CT scans, in the process of fusion. Post-fusion stages may include extracting specific features. As an example in the context of tumor diagnosis, post-fusion analysis may comprise segmenting and isolating areas of focus from the fused picture in order to precisely identify and define anomalies. This post-fusion analysis helps to provide important medical data that may not be obvious from individual photos.

In this study, we used Convolutional Neural Networks, to combine MRI and PET data. This method was to combine data from both methods, which enhanced the overall quality of the combined image. Our findings demonstrate CNNs' usefulness in enhancing multifaceted imaging in medicine for better diagnosis capacities.

1. **Materials and Methods**
2. **Data Collection and Preprocessing**

This study was conducted on a dataset containing previously registered MRI and CT images in nifti format. These images have been normalized to a 79x95x79 resolution atlas. The data set was loaded using the imageDatastore function for processing on MATLAB [1]. During the reading process, the images were first processed using the specially defined script6\_niftiReadCustomized function.

1. **Creating and Training the Convolutional Neural Network (CNN) Model**

Using the MATLAB Deep Learning Toolbox, a convolutional neural network (CNN) model was constructed. Multiple convolutional, ReLU activation, and max pooling layers are included in the model. Lastly, a softmax activation function and a fully linked layer are added to a classification layer.

The data set is split into training and testing sets in order to train the model. Several parameters defined by the trainingOptions function controlled the training process [1] . Specifically, 20 epochs of training were conducted using the stochastic gradient descent (SGD) optimization algorithm.

1. **Evaluation of Classification Results**

Through the use of test set classification results, the trained CNN model's performance was assessed. The model's classification ability and accuracy rate were examined using a confusion matrix.

1. **Feature Maps and Feature Extraction**

Feature maps on the trained CNN model were visualized. Additionally, the deepDreamImage function was used to extract features in a specific layer of the trained model [1].

1. **Results**

* **Ans variable:**

The obtained output provides the following information:

Height of the image: 79 pixels

Width of the image: 95 pixels

This information gives you a basic idea about the dimensions of the images in your work. It indicates that the model you developed was trained and tested on images with these dimensions.

* **Accuracy:**

It represents the accuracy rate of the trained model. When accuracy = 1, it indicates 100% accuracy.

* **Name:**

It represents the name of a specific layer. With name = 'conv', it indicates that the features of a convolutional layer are being analyzed.

* **Iterations:**

The training process comprised a total of 10 iterations. Each iteration represents a stage where the model is trained, and the weights are updated. This indicates that within a specific epoch (the entire dataset being passed through the model), 10 distinct learning steps were taken.

* **Activation Strength:**

In each iteration, activation strength denotes the power of changes on a specific weight set or feature. These values represent the level of focus in the model's learning process. Higher activation strength implies that the updates are more pronounced.

* **Pyramid Level:**

Pyramid level is set to 1 for all iterations. This signifies that updates occur at the most fundamental levels of the model.

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. **Image Plot**

* **CT (Computed Tomography):**

Basic Principle: Computed Tomography is a medical imaging technique that utilizes X-rays to create detailed cross-sectional images inside the body.

Difference: CT images typically show density variations. Different materials with varying densities, such as bones, blood vessels, fluids, and tissues in the brain, appear with different colors and brightness tones in these images.

* **MR (Magnetic Resonance):**

Basic Principle: Magnetic Resonance is an imaging technique that obtains detailed images inside the body using a strong magnetic field and radio waves.

Difference: MR images are generally sensitive to the different water content and molecular structures of tissues. Details in the brain, such as gray and white matter, blood vessels, and edema, may appear with different contrasts and brightness in these images.

metin, ekran görüntüsü, diyagram, renklilik içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. **Training Progress**

This code utilizes the MATLAB Deep Learning Toolbox to train and test a Convolutional Neural Network (CNN) model. The "training progress" window in the output is a tool that visualizes the progress of the training process. Now, let's explain the fundamental components in this output:

* **Accuracy:** This graph shows the accuracy rate on the training and validation datasets. High accuracy rates indicate how well the model fits both the training data and new data.
* **Loss:** The loss graph measures the errors made by the model during training. Loss represents how far the model's predictions are from the true labels. Lower loss values indicate better performance of the model.

Specific to the problem and dataset you are working on, evaluating these graphs allows you to understand the model's performance. If the accuracy is high and the loss is low, it means your model is performing well. Otherwise, you may need to review and adjust parameters or the architecture of your model for improvement.

metin, ekran görüntüsü, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. **Analysis For Trainnetwork Usage (Name: net)**

This output contains a structural analysis of the trained neural network model. It indicates that each numbered layer is a component of the neural network model. Below, we explain the meanings of these layers:

* **imageInputLayer ([79 95 1]):** This is the input layer, processing grayscale images of size 79x95 pixels.
* **convolution2dLayer (5,20):** This is a convolutional layer. It contains 20 filters of size 5x5. These filters are used to extract features from the input image.
* **reluLayer:** This layer includes the Rectified Linear Unit (ReLU) activation function. It takes the results of convolution and introduces non-linearity by setting negative values to zero.
* **maxPooling2dLayer (2, 'Stride', 2):** This layer is a max-pooling layer. It is used to reduce the size of the image and highlight features. The pooling operation uses a window of size 2x2.
* **fullyConnectedLayer (2):** This is a fully connected layer. It contains 2 neurons and represents the output classes. In this example, there are two classes: "tumor present/absent."
* **softmaxLayer:** This layer includes the Softmax activation function. It normalizes the outputs to a range between 0 and 1, providing a probability distribution.
* **classificationLayer:** This is the classification layer. It calculates the cross-entropy loss and produces the classification results.

metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. **True Class & Predicted Class**

This output represents a confusion matrix and classification performance metrics. Now, let's break down this output in detail:

True Class: This represents the actual classes.

Predicted Class: This represents the classes predicted by the model.

H (Hit): This represents correctly classified instances.

T (True): This represents the true class.

1 (One): This indicates that the classification is related to the "tumor present" class.

Confusion Matrix: This is a matrix containing the true and predicted classes for each class. The "1" values in the top-left and bottom-right indicate that the "tumor present" class is correctly classified (%100 accuracy).

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. **Layer Conv Features**

This code utilizes MATLAB Deep Learning Toolbox to train and test a Convolutional Neural Network (CNN) model. The output Layer Conv Features displays the feature maps obtained from a convolutional layer of the CNN. Each feature map corresponds to the features extracted by filters in the preceding convolutional layer from the input image.

The black and white icons in the Layer Conv Features output represent that each feature map captures a specific feature or pattern from the input image. These feature maps highlight significant features and objects learned by the network. Such visualizations can be used to understand the learning process of the model and visually observe which features the model is focusing on.

ekran görüntüsü, metin, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

# **References**

|  |  |
| --- | --- |
| [1] | The MathWorks, Inc., 1994. [Çevrimiçi]. Available: https://www.mathworks.com/. |