



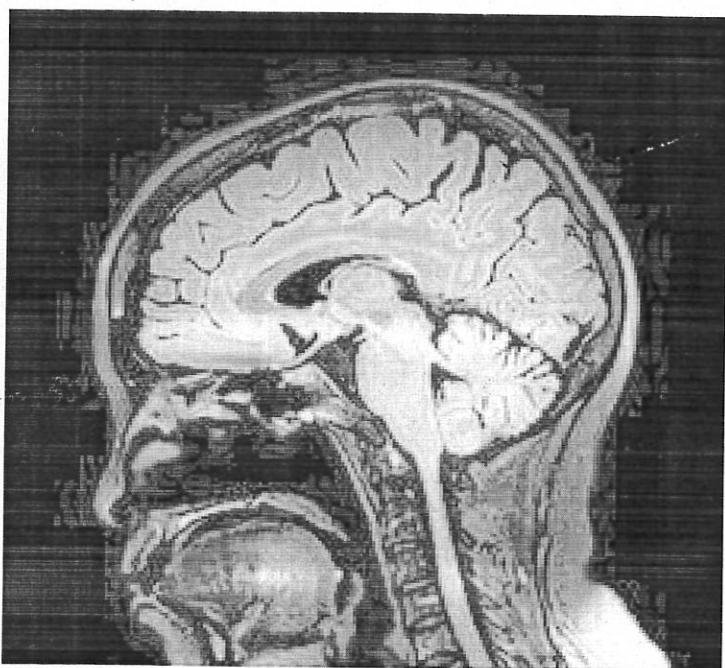
MCKV INSTITUTE OF ENGINEERING
Liluah, Howrah West Bengal

CERTIFICATE

This is to certify that **NITISH KUMAR** Of Information Technology Course in Bachelors of Technology has successfully completed their project work on
(Automation of Denoising of ICA components in FSL using FIX)

In the partial fulfillment of requirement as prescribed by the NMR at INMAS, DRDO, in the period June 25th-July 26th 2019. Guided by Dr. Shilpi Modi, Scientist 'E', INMAS, DRDO.

Dr Shilpi Modi
Scientist 'E'
Department of NMR
INMAS , DRDO



*Automation of Denoising of
ICA
Components in FSL using FIX*

ACKNOWLEDGEMENT

Our experience of project at INMAS,DRDO has been completed successfully, thanks to the support staff of many friends & colleagues with gratitude. We wish to acknowledge all of them. However, we wish to make special mention of the following.

First of all we are thankful of our project mentor Dr. Shilpi Modi ma'am under whose guideline we were able to complete our project. We are wholeheartedly thankful to her for giving us her valuable time and attention and for providing us a systematic way for completing our project in time.

We must make special mention of Mr Deepak Sharma sir, our project in charge for his co-operation & assistance in solving technical problems. We would thank to our all lab maintenance staff for providing us assistance in various h/w & s/w problem encountered during course of our project.

We are also very thankful to respected Dr Indrajit De sir and Mr. Naveen Kumar Soni sir who gave us an opportunity to present this project.

Participants

TABLE OF CONTENTS

1. INTRODUCTION

- Introduction to Project
 - Introduction to language
-

2. SYSTEM CONFIRURATION

- Tools required
- Installation Guide

3. A SIMPLE GUI FOR FIX

- Code
- How to use it

4. Analysis & Results

- Steps for running FIX
- Results

5. ERRORS HANDLING

- Common Errors

6. CONCLUSION

7. BIBLIOGRAPHY

1. INTRODUCTION

Utilitarian attractive reverberation imaging or useful MRI (fMRI) measures cerebrum action by distinguishing changes related with blood stream. This strategy depends on the way that cerebral blood stream and neuronal enactment are coupled. At the point when a territory of the mind is being used, blood stream to that area additionally increments.

The essential type of fMRI utilizes the blood-oxygen-level ward (BOLD) differentiate, found by Seiji Ogawa. This is a kind of specific mind and body sweep used to delineate movement in the cerebrum or spinal rope of people or different creatures by imaging the adjustment in blood stream (hemodynamic reaction) identified with vitality use by synapses. Since the mid 1990s, fMRI has come to command cerebrum mapping research since it doesn't expect individuals to experience shots or medical procedure, to ingest substances, or to be presented to ionizing radiation. This measure is every now and again ruined by clamor from different sources; thus, factual systems are utilized to extricate the basic sign. The subsequent mind actuation can be graphically spoken to by shading coding the quality of enactment over the cerebrum or the particular locale contemplated. The procedure can confine action to inside millimeters be that as it may, utilizing standard strategies, no superior to inside a window of a couple of moments. Different techniques for acquiring complexity are blood vessel turn naming and dissemination MRI. The last system is like BOLD fMRI however gives complexity dependent on the extent of dissemination of water atoms in the cerebrum.

Notwithstanding recognizing BOLD reactions from action because of assignments/improvements, the fMRI worldview incorporates likewise resting state fMRI, or taskless fMRI, which estimates the subjects' standard BOLD change. Since around 1998, this line of studies has uncovered the presence and properties of the default mode arrange (DMN), otherwise known as 'Resting State Network' (RSN), a practically associated neural system of obvious 'cerebrum states'.

fMRI is utilized both in the examination world, and to a lesser degree, in the clinical world. It can likewise be joined and supplemented with different proportions of cerebrum physiology, for example, EEG and NIRS. More up to date strategies which improve both spatial and time goals are being looked into, and these to a great extent use biomarkers other than the BOLD sign. A few organizations have created business items, for example, lie locators dependent on fMRI systems, however the exploration isn't accepted to be ready enough for across the board commercialization.

2. SYSTEM CONFIGURATION

Tools Required :-

- FSL
- R & R STUDIO
- MATLAB
- FIX
- PYTHON

1. FSL Installation Steps

- Download latest version of FSL (6.0.1) form here. [Link](#)
- Install Python version 2.7 on system.
- Open the downloaded path of the file from the terminal and type:
cd ~/Downloads :
python fslinstaller.py
- After installation type : nano~/.bashrc and paste the following:
FSLDIR=/usr/local/fsl
. \${FSLDIR}/etc/fslconf/fsl.sh
PATH=\${FSLDIR}/bin:\${PATH}
export FSLDIR PATH

2. R & R Studio Installation Steps

- Install R(3.6.1 or >=3.0) just by typing R in the terminal and get the installation command
- Download .deb file from [link](#). Open it and install R Studio.
- After installation open the R Studio and install the following packages required for FIX:
 - ‘kernlab’ version 0.9.24
 - ‘ROCR’ version 1.0.7
 - ‘class’ version 7.3.14
 - ‘party’ version 1.0.25
 - ‘e1071’ version 1.6.7
 - ‘randomForest’ version 4.6.12

Some additional packages :-

1. Sandwich
2. Coin
3. Zoo
4. Survival
5. Modeltools
6. Mvtnorm

- Installation can be done by just typing : `install.packages("package Name")` in the R Studio.

3. MATLAB Installation Steps

- Download MATLAB for Linux form [Link](#) and unzip the folder after downloading it.
- Read the installation guide for installing it on system.
- Install following toolboxes:
 - Statistics
 - Signal Processing

4. FIX Installation Steps

- Download FIX setup form [link](#) and unzip the folder in the `/usr/local/`.

5. Configure FIX

- Open terminal and go to fix directory :
`cd /usr/local/fix` and type :
 1. `cat settings.sh`
 2. `./settings.sh`
 3. `Source settings.sh`
- Check if configure or not by :
 1. `echo $FSL_FIX_R-CMD` //it will give `/usr/local/R`
 2. `echo$FSL_FIX_MCRROOT`
`//opt/fmrib/MATLAB_Complier_Runtime`

3.FIX_GUI

Codes:-

```
import Tkinter, tkFileDialog, Tkconstants ,os
import os.path
import sys
import re
from Tkinter import *

#Creating Font Styles

font10 = "-family {Segoe UI} -size 12 -weight normal -slant " \
    "roman -underline 0 -overstrike 0"
font11 = "-family {Segoe UI} -size 14 -weight normal -slant " \
    "roman -underline 0 -overstrike 0"
font13 = "-family {Segoe UI} -size 13 -weight normal -slant " \
    "roman -underline 0 -overstrike 0"
font15 = "-family {Courier New} -size 14 -weight normal -slant" \
    " roman -underline 0 -overstrike 0"
font9 = "-family {Segoe UI} -size 14 -weight bold -slant roman" \
    " -underline 0 -overstrike 0"
font12 = "-family {Segoe UI} -size 13 -weight bold -slant " \
    "roman -underline 0 -overstrike 0"

# Defining Functions

def browsefunc():
    global folder_path
    filename = tkFileDialog.askdirectory(parent=root, initialdir='/home/mac/Desktop', title='Browse .feat
    folder directory')
    Entry2.config(text=filename)# take input in this function and return as string.
    folder_path.set(filename)
    os.chdir(filename)
    cwd=os.getcwd()
    print("Getting .feat Directory .....\\n")
```

```

print("Feat Directory :- "+ cwd +"\n")
return filename

# Defining Function for Running FIX

def fixfunc():
    global thres_val
    thres_val=Entry1.get()
    path=os.path.join('/usr/local/fix/fix'+ " "+str(folder_path.get())+ " "
    +' /usr/local/fix/training_files/Standard.RData'+" "+thres_val)
    os.system(path)
    denoisingfunc() # Calls Denoised Function

def denoisingfunc():
    mylines = [] # Declare an empty list named mylines.
    with open ('.fix', 'rt') as myfile: # Open lorem.txt for reading text data.
        for myline in myfile: # For each line, stored as myline,
            mylines.append(myline.rstrip('\n')) # add its contents to mylines.
    mynewlist=" ".join(map(str,mylines))
    mylist=[str(mynewlist).replace(' ','') for myline in mylines]
    lis=" ".join(mylist)
    denoised_dir_path=os.path.join('fsl_refilt -i filtered_func_data -o denoised_data -d
    filtered_func_data.ica/melodic_mix -f'+ " "+"""+lis+""")
    print("Fix Completed \n Now Denoising ")
    print(denoised_dir_path)
    os.system(denoised_dir_path)
    print("successfully") # Denoising Completed

# Defining Function for Running Melodic

def melodic_denoisingfunc():
    global thres_val
    thres_val=Entry1.get()
    mel_path=str(folder_path.get())+"/denoised_data.ica"
    os.chdir(mel_path)
    print(os.getcwd()+"\n")
    mylines =[] # Declare an empty list named mylines.

```

```

with open ('.fix', 'rt') as myfile: # Open lorem.txt for reading text data.
    for myline in myfile:          # For each line, stored as myline,
        mylines.append(myline.rstrip('\n'))      # add its contents to mylines.
mynewlist=".join(map(str,mylines))
mylist=[str(mynewlist).replace(' ','') for myline in mylines]
lis=" ".join(mylist)

mel_denoised_dir_path=os.path.join('fsl_refilt -i filtered_func_data -o denoised_data -d
filtered_func_data.ica/melodic_mix -f'+ " "+lis+"")
f=open('.fix')
f1=open('output.txt','a')
for x in f.readlines():
    f1.write(x)
    f1.write("\n")
print("copy done\n")
print("Fix Completed \n Now Denoising ")
print(mel_denoised_dir_path+"\n")
os.system(mel_denoised_dir_path)
print("successfully")

def melodifunc():
    global thres_val
    thres_val=Entry1.get()
    melo_path=os.path.join('/usr/local/fix/fix'+ " "+str(folder_path.get())+"/denoised_data.ica"+"
'+"/usr/local/fix/training_files/Standard.RData "+str(thres_val))
    print("Start Denoising on melodic Directory .....\\n")
    copy_dir_path=os.path.join('cp -avr'+ " "+str(folder_path.get())+"/mc"+
"+str(folder_path.get())+"/denoised_data.ica")
    if(os.path.exists(str(folder_path.get())+"/mc")):
        print("\n")
    else:
        os.system(copy_dir_path)
    print(melo_path+"\n")
    os.system(melo_path)
    melodic_denoisingfunc()

# Exit Button

```

```
def exit():
    root.destroy() #Removes the hidden root window
    sys.exit() #Ends the script

# Creating Frame Window

root=Tk()
root.title("FIX_GUI")
thres_val=int()
folder_path=StringVar()
root.geometry("595x250+205+101")
root.configure(background="#EEE8AA")

Labelframe1 = LabelFrame(root)
Labelframe1.place(relx=0.06, rely=0.040, relheight=0.803
                  , relwidth=0.908)
Labelframe1.configure(relief='groove')
Labelframe1.configure(borderwidth="4")
Labelframe1.configure(font=font9)
Labelframe1.configure(foreground="black")
Labelframe1.configure(background="#F5F5DC")
Labelframe1.configure(width=600)

browsebutton = Button(root,command=browsefunc)
browsebutton.place(relx=0.807, rely=0.076, height=44, width=87)
browsebutton.configure(activebackground="#ececce",activeforeground="#000000",background="#FF63
47")
browsebutton.configure(borderwidth="0",disabledforeground="#a3a3a3",font=font12,foreground="#00
0000")
browsebutton.configure(highlightbackground="#d9d9d9",highlightcolor="black",text="Browse")

Entry2 = Label(root)
Entry2.place(relx=0.073, rely=0.076,height=40, relwidth=0.723)
Entry2.configure(background="white")
Entry2.configure(disabledforeground="#a3a3a3")
Entry2.configure(font=font11)
Entry2.configure(foreground="#000000")
```

```
Entry2.configure(width=394)

# Threshold label

Label2 = Label(Labelframe1)
Label2.place(relx=0.233, rely=0.300, height=41, width=160
            , bordermode='ignore')
Label2.configure(background="#F5F5DC")
Label2.configure(disabledforeground="#a3a3a3")
Label2.configure(font=font13)
Label2.configure(foreground="#000000")
Label2.configure(text="Threshold Value")
Label2.configure(width=160)
```

Take Threshold value by user

```
Entry1 = Entry(Labelframe1)
Entry1.place(relx=0.550, rely=0.300, height=42, relwidth=0.090
            , bordermode='ignore')
Entry1.configure(background="white")
Entry1.configure(disabledforeground="#a3a3a3")
Entry1.configure(font=font13)
Entry1.configure(foreground="#000000")
Entry1.configure(insertbackground="black")
Entry1.configure(width=44)
Entry1.insert(0, 20)
```

Creating Feat Button to run FIX

```
Button2 = Button(Labelframe1,command=fixfunc)
Button2.place(relx=0.084, rely=0.703, height=44, width=157)
Button2.configure(activebackground="#ececec")
Button2.configure(activeforeground="#000000")
Button2.configure(background="#d9d9d9")
Button2.configure(disabledforeground="#a3a3a3")
Button2.configure(font=font11)
Button2.configure(foreground="#000000")
```

```
Button2.configure(highlightbackground="#d9d9d9")
Button2.configure(highlightcolor="black")
Button2.configure(pady="0")
Button2.configure(text="Run FIX on Feat")
Button2.configure(width=157)

# Creating Melodic Button to run FIX

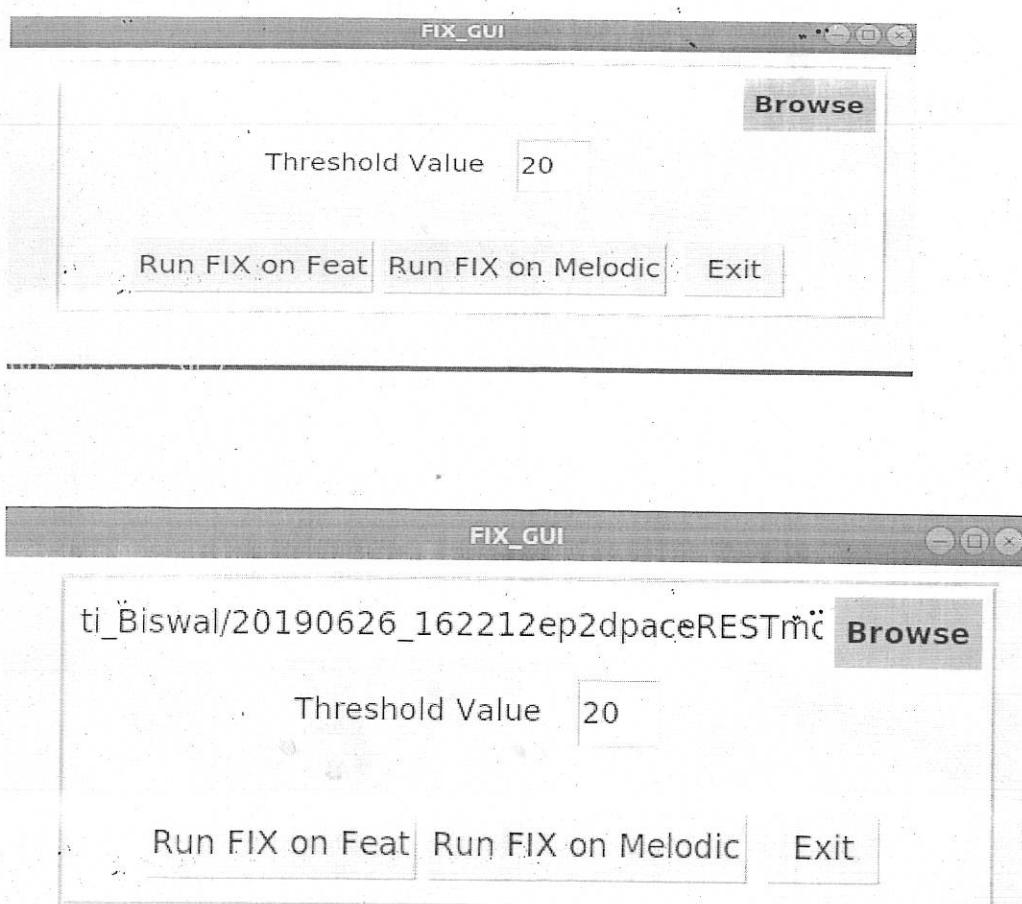
Button3 = Button(Labelframe1,command=melodifunc)
Button3.place(relx=0.387, rely=0.703, height=44, width=187)
Button3.configure(activebackground="#ececce")
Button3.configure(activeforeground="#000000")
Button3.configure(background="#d9d9d9")
Button3.configure(disabledforeground="#a3a3a3")
Button3.configure(font=font11)
Button3.configure(foreground="#000000")
Button3.configure(highlightbackground="#d9d9d9")
Button3.configure(highlightcolor="black")
Button3.configure(pady="0")
Button3.configure(text="Run FIX on Melodic")
Button3.configure(width=187)

# Creating Exit Button

Button4 = Button(Labelframe1,command=exit)
Button4.place(relx=0.756, rely=0.703, height=44, width=67)
Button4.configure(activebackground="#ececce")
Button4.configure(activeforeground="#000000")
Button4.configure(background="#d9d9d9")
Button4.configure(disabledforeground="#a3a3a3")
Button4.configure(font=font11)
Button4.configure(foreground="#000000")
Button4.configure(highlightbackground="#d9d9d9")
Button4.configure(highlightcolor="black")
Button4.configure(pady="0")
Button4.configure(text="Exit")
Button4.configure(width=67)
root.mainloop()
```

How to use GUI Interface

1. Browse the .feat directory.
2. Click on “Run FIX on feat” button for running and denoising on .feat directory
3. Click on “Run FIX on Melodic” button for running and denoising on denoised_data.ica directory or melodic output directory.
4. Threshold value is set to 20 by default ,it can be varied



Some Important points before running FIX

» Make sure that FSL(latest Version),R(3.6.1 or >=3.0),R packages (same version) as in FSL FIX User Guide & Matlab.

» Before running fix Make sure that it is configured it correctly.

» If denoising data for the first time after running feat then while running fix on it give .feat directory in place of mel.ica not the melodic directory.

/usr/local/fix/fix<mel.ica>/usr/local/fix/training_files/Standard.RData 20

» 20 refers to the thresholding of good vs bad components,sensible values are generally in the range of 5-20.

» We have already tested by varying the threshold value form 5-30 results approx same. As threshold value increases more and more noise components are found.

» If denoising data for the second time after running Melodic then while running fix on it give denoised_data.ica directory in place of mel.ica i.e the melodic output directory.

/usr/local/fix/fix <mel.ica> /usr/local/fix/training_files/Standard.RData 20

Steps for running FIX

After Feat analysis :-

1. /usr/local/fix/fix <mel.ica> /usr/local/fix/training_files/Standard.RData 20
 - <mel.ica> is path to feat output directory i.e .feat directory
2. fsl_refilt -i filtered_func_data -o denoised_data -d filtered_func_data.ica/melodic_mix -f 'pass components that are considered as noise separated by comma'
eg. fsl_refilt -i filtered_func_data -o denoised_data -d filtered_func_data.ica/melodic_mix -f "1,2,3,6,8,9,10,11,12,13,14,15,18,20,22,23,24,28,29,31,32,33,34,37,38,39,40,41,43,44,45,47,48,49,50"

On successful run, it will generate .fix file (components that are considered as noise to be removed) and with a cleaned version of the 4D preprocessed FMRI

After Melodic analysis :-

1. Just copy .../.feat/mc folder to .../.feat/denoised_data.ica for correct output
2. Run /usr/local/fix/fix <mel.ica> /usr/local/fix/training_files/Standard.RData 20
 - <mel.ica> is path to melodic output directory i.e denoised_data.ica directory
3. fsl_refilt -i filtered_func_data -o denoised_data -d filtered_func_data.ica/melodic_mix -f 'pass components that are considered as noise separated by comma'

On successful run, it will generate .fix file (components that are considered as noise to be removed) and with a cleaned version of the 4D preprocessed FMRI data: filtered_func_data_clean.nii.gz.

Results

Feat Analysis:-

```
Activities Tk Thu 14:27
mac@mac-ThinkCentre-M910z:~/Desktop$ clear

mac@mac-ThinkCentre-M910z:~/Desktop$ python New_Fix_GUI.py
Getting .feat Directory ......

Feat Directory -- /home/mac/Desktop/NMR_FIX_DATA/Sanskruti_Biswal/20190626_162212ep2dpacerESTnocodypy2NVSANSKRUTIBISWAL010a001.feat

Start Denoising on Feat Directory ......

FIX Feature extraction for Melodic output directory: /home/mac/Desktop/NMR_FIX_DATA/Sanskruti_Biswal/20190626_162212ep2dpacerESTnocodypy2NVS
create edge masks
run FAST
registration of standard space masks
extract features
FIX Classifying components in Melodic directory: /home/mac/Desktop/NMR_FIX_DATA/Sanskruti_Biswal/20190626_162212ep2dpacerESTnocodypy2NVSANSKRUTIBISWAL010a001.feat using training file: /usr/local/flx/training_f
iles/Standard.Rdata and threshold 20
FIX Applying cleanup using cleanup file: /home/mac/Desktop/NMR_FIX_DATA/Sanskruti_Biswal/20190626_162212ep2dpacerESTnocodypy2NVSANSKRUTIBISWAL010a001.feat/fix4melview_Standard_thr20.txt and motion cleanup set to 0
Fix Completed
Now Denoising
fsf_refilt -i filtered_func_data -o denoised_data -d filtered_func_data.lca/melodic_mix -f '1,2,3,4,6,8,9,11,12,13,16,19,22,23,24,25,26,29,32,33,37,38,39,41,42,44,45,47,48,49,54,55,56,58,59'
successfully
```

Melodic Analysis:-

- Components that are considered as noise after running fix on feat directory:
 [1,2,3,4,6,8,9,11,12,13,16,19,22,23,24,25,26,29,32,33,37,38,39,41,42,44,45,47,48,49,54,55,56,58,59]
- Components that are considered as noise by manually:
 [1,2,3,4,6,8,9,11,12,13,16,19,22,23,24,25,26,32,33,37,38,39,41,42,44,45,47,48,49,54,55,56,57,58,59]
- Components that are considered as noise by after running fix on Melodic directory
 [27,28,29,30,32,33,35]
- Components that are considered as noise by manually:
 [2,12,27,28,29,30,31,32,33,34,35]

COMMON ERRORS

- » Error at terminal "No valid labelling file specified "
- » After running try to see the following log files:
 - 1.<subject.ica>/fix/logMatlab.txt (this should show errors in Matlab part, i.e. features extraction)
 - 2.<subject.ica>/.fix.log
 - 3.<subject.ica>/.fix_2b_predict.log (those are log file in general for the whole routine)

You'll probably find errors related to Matlab or R, so you might need to check your settings.sh file following the setup instructions described in the FIX README file

Ways to resolve these errors :-

 1. if there is error in fix_2b_predict.log then there is R & its packages version problem(try to install same packages)
 2. if there is error in logMatlab.txt then check matlab configuration i.e FSL_FIX_MATLAB_ROOT or it is settings.sh configuration problem .
 3. if getting error like features.csv file not found or prefiltered_func_data_mcf.mat not found means
 .../feat/mc folder is not present in the directory.

CONCLUSION

- ✓ FIX has helped to enable the work faster and in more efficient manner.
- ✓ The caution needs to be taken at the threshold value entered in the application as 20, some problems are seen in increasing or decreasing the threshold value.
- ✓ The setup needs the mentioned versions of the software or upgraded ones only.
- ✓ Some subjects might not function properly in FIX and may not give the desired output as FEAT might not have been successfully done.

BIBLIOGRAPHY

- <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FIX/FAQ>
- <https://www.jiscmail.ac.uk/cgi-bin/webadmin?A2=FSL;166450a3.1412>
- https://github.com/jelman/FSL_FIX/blob/master/README