

## Riconoscimento della patologia con CNN

```
clc;
clear all;

%creiamo un database
imagespathCovid = dir('C:\Users\Virginia\Desktop\Ingegneria Biomedica\SEGNALI\PROGETTO MERONE\');
imagespathNormal = dir('C:\Users\Virginia\Desktop\Ingegneria Biomedica\SEGNALI\PROGETTO MERONE\');
imagespathViralPneumonia = dir('C:\Users\Virginia\Desktop\Ingegneria Biomedica\SEGNALI\PROGETTO MERONE\');

%elimino le prime righe
ImagesPathCovid = imagespathCovid(4:end,:);
ImagesPathNormal = imagespathNormal(3:end,:);
ImagesPathViralPneumonia = imagespathViralPneumonia(3:end,:);

%definisco il numero di immagini che ci sono in ogni cartella
dimCovid = size(ImagesPathCovid,1);
dimNormal = size(ImagesPathNormal,1);
dimViral = size(ImagesPathViralPneumonia,1);
images = struct; %creo una struct senza fields
```

```
for i=1:dimCovid+dimNormal+dimViral
    if i<=dimCovid %associamo a 1 tutte le immagini Covid
        images(i).images=im2gray(imread([ImagesPathCovid(i).folder, '/',ImagesPathCovid(i).name]));
        images(i).label=categorical(1);
        images(i).images=imresize(images(i).images,[350 400]);

        elseif i<=dimCovid+dimNormal %associo a 2 le immagini Normal
            images(i).images=im2gray(imread([ImagesPathNormal(i-dimCovid).folder, '/',ImagesPathNormal(i-dimCovid).name]));
            images(i).label=categorical(2);
            images(i).images=imresize(images(i).images,[350 400]);

            else %3 = immagini Viral Pneumonia
                images(i).images=im2gray(imread([ImagesPathViralPneumonia(i-dimCovid-dimNormal).folder, '/',ImagesPathViralPneumonia(i-dimCovid-dimNormal).name]));
                images(i).label=categorical(3);
                images(i).images=imresize(images(i).images,[350 400]);
            end
        end
end
```

```
Accuracy=[];
accuracy=[];
Precision=[];
Recall=[];
%3 fold cross validation -> 3 perché abbiamo 3 cartelle
for i=1:3
    Random=images(randperm(size(images,2))); %randomizzo le immagini
    Training= Random(1:250);
    Test=Random(251:282);
    Validation=Random(283:312);
    training=struct2table(Training); %convertito in tabella
    test=struct2table(Test);
    validation=struct2table(Validation);
```

```

for j=2:1:12 %creo la rete, tutta la struttura che costruiamo
    layers=[
        imageInputLayer([350 400 1]) %grandezza immagini 350x400x1 1=C--> immagine gray, c=3 --> RGB
        convolution2dLayer(5,20, 'Stride',j) %kernel 5x5 e ne prendiamo 20, Stride é di quanto si sposta
        reluLayer %retta lineare per x>=0
        maxPooling2dLayer(2) %maxPooling 2x2
        fullyConnectedLayer(3)
        softmaxLayer %normalizza ogni uscita con le altre
        classificationLayer]; %risultato softmax a quale neurone corrisponde

    options= trainingOptions('adam', 'MaxEpochs',8,'InitialLearnRate', 1e-4,'Verbose', false,
    net=trainNetwork(training,layers,options); %addestro la rete andando a passare i dati di training
    YPred=classify(net,validation); %vettore di predizione per le validation
    ConfusionMat=confusionmat(validation.label,YPred); %matrice di confusione - valuto prestazioni
    accuracy(j)=(sum(diag(ConfusionMat))/30)*100; %accuratezza attraverso matrice di confusione

end

[M,I]=max(accuracy); %livello di stride associato alla max accuracy
layers=[...
    imageInputLayer([350 400 1])
    convolution2dLayer(5,20, 'Stride',I) %livello stride I, ricavato dal ciclo for precedente
    reluLayer
    maxPooling2dLayer(2)
    fullyConnectedLayer(3)
    softmaxLayer
    classificationLayer];
options= trainingOptions('adam','MaxEpochs',8,'InitialLearnRate', 1e-4, 'Verbose', false, 'MinimizeCost',true);
net=trainNetwork(training,layers,options);
Prediction=classify(net,test);
ConfusionMat=confusionmat(test.label,Prediction);
Accuracy(i)=(sum(diag(ConfusionMat))/32)*100;
Precision(i,:)=[(ConfusionMat(1,1)/sum(ConfusionMat(:,1))) (ConfusionMat(2,2)/sum(ConfusionMat(:,2)))];
Recall(i,:)=[(ConfusionMat(1,1)/sum(ConfusionMat(1,:))) (ConfusionMat(2,2)/sum(ConfusionMat(2,:)))];

end

```



