任务 1：

计算每个字母的概率，分别取最大值为结果

```
numChar=length(images);
pred=zeros([numChar,1]);
K=imageModel.K;
for n=1:numChar
    p=ones([1,K]);
    for k=1:K-1
        p(k)=exp(imageModel.W(k,:)*images(n).img(:)+imageModel.bias(k));
    end
    p=p./sum(p);
    [~,pred(n)]=max(p);
end
```

任务 2：

1）计算每个字母的识别结果概率，并联合两个字母的连续概率，使用 ICM 进行迭代求最大值。使用任务 1 中的结果作为迭代起始值。

```
numChar=length(images);
K=imageModel.K;
pred=randi(K,[numChar,1]);


p1=ones([numChar,K]);

for n=1:numChar
    for k=1:K-1
        p1(n,k)=exp(imageModel.W(k,:)*images(n).img(:)+imageModel.bias(k));
    end
    p1(n,:)=p1(n,:)./sum(p1(n,:));
end

[~,pred]=max(p1,[],2);
if numChar<3
    return
end
max_p=zeros([numChar,1]);

while 1
    for n=1:numChar
        p=zeros([K,1]);
        for k =1:K
            pred(n)=k;
            p(k)=log(p1(numChar,pred(numChar)));
            for w=1:numChar-1   %计算单词条件概率
                p(k)=p(k)+log(p1(w,pred(w)))+log(pairwiseModel(pred(w),pred(w+1)));
            end
        end
        [max_p(n),pred(n)]=max(p);
    end
    if all (~(diff(max_p)))
        break
    end
end
```

2）使用维特比算法求解最大值

```
numChar=length(images);
pred=zeros([numChar,1]);
K=imageModel.K;

p1=ones([numChar,K]);

for n=1:numChar
    for k=1:K-1
        p1(n,k)=exp(imageModel.W(k,:)*images(n).img(:)+imageModel.bias(k));
    end
    p1(n,:)=p1(n,:)./sum(p1(n,:));
end

if numChar<3
    return
end

pred_m=zeros([K,numChar]);
for n=1:K
    pred_m(n,1)=n;
end
p_m=log(p1(1,:));

for n=2:numChar
    for k_m =1:K
        p=zeros([K,1]);
        for k=1:K
            p(k)=log(p1(n,k))+log(pairwiseModel(pred_m(k_m,n-1),k));
        end
        [P,M]=max(p);
        p_m(k_m)=p_m(k_m)+P;
        pred_m(k_m,n)=M;
    end
end
[~,M]=max(p_m);
pred=pred_m(M,:);
```

任务 3：

1）计算每个字母的识别结果概率，并联合两个字母和三个字母的连续概率，使用 ICM 进行迭代求最大值。使用任务 1 中的结果作为迭代起始值。

```
numChar=length(images);
K=imageModel.K;
pred=randi(K,[numChar,1]);

p1=ones([numChar,K]);

for n=1:numChar
    for k=1:K-1
        p1(n,k)=exp(imageModel.W(k,:)*images(n).img(:)+imageModel.bias(k));
    end
    p1(n,:)=p1(n,:)./sum(p1(n,:));
end

[~,pred]=max(p1,[],2);  %迭代初始值
if numChar<4
    return
end
max_p=zeros([numChar,1]);

while 1
    for n=1:numChar
        p=zeros([K,1]);
        for k =1:K
            pred(n)=k;
            p(k)=log(p1(numChar,pred(numChar)))+log(p1(numChar-1,pred(numChar-1)))+log(pairwiseModel(pred(numChar-1),pred(numChar)));
            for w=1:numChar-2   %计算单词条件概率
                p(k)=p(k)+log(p1(w,pred(w)))+log(pairwiseModel(pred(w),pred(w+1)))+log(tripletModel(pred(w),pred(w+1),pred(w+2)));
            end
        end
        [max_p(n),pred(n)]=max(p);
    end
    if all (~(diff(max_p)))
        break
    end
end
```

2）使用任务 2 中维特比算法求得结果作为初始值。

```
numChar=length(images);
K=imageModel.K;
pred=randi(K,[numChar,1]);

p1=ones([numChar,K]);

for n=1:numChar
    for k=1:K-1
        p1(n,k)=exp(imageModel.W(k,:)*images(n).img(:)+imageModel.bias(k));
    end
    p1(n,:)=p1(n,:)./sum(p1(n,:));
end

pred = RunInference2w(images, imageModel, pairwiseModel, tripletModel); %迭代初始值

if numChar<4
    return
end
max_p=zeros([numChar,1]);

while 1
    for n=1:numChar
        p=zeros([K,1]);
        for k =1:K
            pred(n)=k;
            p(k)=log(p1(numChar,pred(numChar)))+log(p1(numChar-1,pred(numChar-1)))+log(pairwiseModel(pred(numChar-1),pred(numChar)));
            for w=1:numChar-2   %计算单词条件概率
                p(k)=p(k)+log(p1(w,pred(w)))+log(pairwiseModel(pred(w),pred(w+1)))+log(tripletModel(pred(w),pred(w+1),pred(w+2)));
            end
        end
        [max_p(n),pred(n)]=max(p);
    end
    if all (~(diff(max_p)))
        break
    end
end
```

任务 4：

1）计算每个字母的识别结果概率，并联合两个字母和三个字母的连续概率、以及图像相似度最大的两个图像，使用 ICM 进行迭代求最大值。使用任务 1 中的结果作为迭代起始值。

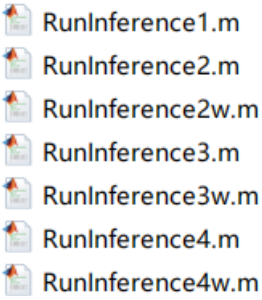
```
numChar=length(images);
K=imageModel.K;
pred=randi(K,[numChar,1]);

p1=ones([numChar,K]);

for n=1:numChar
    for k=1:K-1
        p1(n,k)=exp(imageModel.W(k,:)*images(n).img(:)+imageModel.bias(k));
    end
    p1(n,:)=p1(n,:)./sum(p1(n,:));
end

[~,pred]=max(p1,[],2);  %
if numChar<4
    return
end
max_p=zeros([numChar,1]);

while 1
    for n=1:numChar
        p=zeros([K,1]);
        for k =1:K
            pred(n)=k;
            p(k)=log(p1(numChar,pred(numChar)))+log(p1(numChar-1,pred(numChar-1)))+log(pairwiseModel(pred(numChar-1),pred(numChar)));
            for w=1:numChar-2   %计算单词条件概率
                p(k)=p(k)+log(p1(w,pred(w)))+log(pairwiseModel(pred(w),pred(w+1)))+log(tripletModel(pred(w),pred(w+1),pred(w+2)));
            end
            c=zeros([numChar*(numChar-1)/2,1]);
            c0=1;
            for i =1:numChar-1
                for j =i+1:numChar
                    if pred(i)~=pred(j)
                        c(c0)=1;
                    else
                        c(c0)=ImageSimilarity(images(i).img,images(j).img);
                    end
                    c0=c0+1;
                end
            end
            c=sort(c);
            p(k)=p(k)+log(c(length(c)))+log(c(length(c)-1));
        end
        [max_p(n),pred(n)]=max(p);
    end
    if all (~(diff(max_p)))
        break
    end
end
```

2）使用任务 2 中维特比算法求得结果作为初始值。

```
numChar=length(images);
K=imageModel.K;
pred=randi(K,[numChar,1]);

p1=ones([numChar,K]);

for n=1:numChar
    for k=1:K-1
        p1(n,k)=exp(imageModel.W(k,:)*images(n).img(:)+imageModel.bias(k));
    end
    p1(n,:)=p1(n,:)./sum(p1(n,:));
end

pred = RunInference2w(images, imageModel, pairwiseModel, tripletModel);
if numChar<4
    return
end
max_p=zeros([numChar,1]);

while 1
    for n=1:numChar
        p=zeros([K,1]);
        for k =1:K
            pred(n)=k;
            p(k)=log(p1(numChar,pred(numChar)))+log(p1(numChar-1,pred(numChar-1)))+log(pairwiseModel(pred(numChar-1),pred(numChar)));
            for w=1:numChar-2   %计算单词条件概率
                p(k)=p(k)+log(p1(w,pred(w)))+log(pairwiseModel(pred(w),pred(w+1)))+log(tripletModel(pred(w),pred(w+1),pred(w+2)));
            end
            c=zeros([numChar*(numChar-1)/2,1]);
            c0=1;
            for i =1:numChar-1
                for j =i+1:numChar
                    if pred(i)~=pred(j)
                        c(c0)=1;
                    else
                        c(c0)=ImageSimilarity(images(i).img,images(j).img);
                    end
                    c0=c0+1;
                end
            end
            c=sort(c);
            p(k)=p(k)+log(c(length(c)))+log(c(length(c)-1));
        end
        [max_p(n),pred(n)]=max(p);
    end
    if all (~(diff(max_p)))
        break
    end
end
```

结果分析：

不同的算法分别存在不同的函数中，其中带有 w 的函数为使用维特比算法或使用维特比路径作为初始值进行 ICM 迭代。



通过修改 wordPredictions.m 中使用的识别函数，可以得到不同的结果。

```
numWords = length(allWords);
wordPredictions = cell(numWords, 1);

for i = 1:numWords
    wordPredictions{i} = RunInference4w(allWords{i}, imageModel, pairwiseModel, tripletModel);
end

end
```

运行的结果如下所示：

	ICM	维特比
任务 1	530 / 691 characters (76.70% accuracy) 22 / 100 words (22.00% accuracy)	-
任务 2	529 / 691 characters (76.56% accuracy) 25 / 100 words (25.00% accuracy)	531 / 691 characters (76.85% accuracy) 28 / 100 words (28.00% accuracy)
任务 3	535 / 691 characters (77.42% accuracy) 30 / 100 words (30.00% accuracy)	547 / 691 characters (79.16% accuracy 34 / 100 words (34.00% accuracy)
任务 4	540 / 691 characters (78.15% accuracy) 32 / 100 words (32.00% accuracy)	548 / 691 characters (79.31% accuracy 35 / 100 words (35.00% accuracy)