基于深度学习的方法实现透过散射介质成像

答辩人：韦璐 | 导师：王自强

少年班学院 光学与光学工程系

2021-06-08

中国科学技术大学

University of Science and Technology of China

概　要

选题动机和研究过程

散斑图像基本原理

深度学习图像处理

基于深度学习的散斑图像处理

总结和展望

选题动机

- 光学系统图像处理问题在许多问题中非常重要
- 散斑图像光学系统中非常普遍
- 神经网络非常适合处理图像
- 利用神经网络处理散斑图像可以克服许多传统方法的缺点

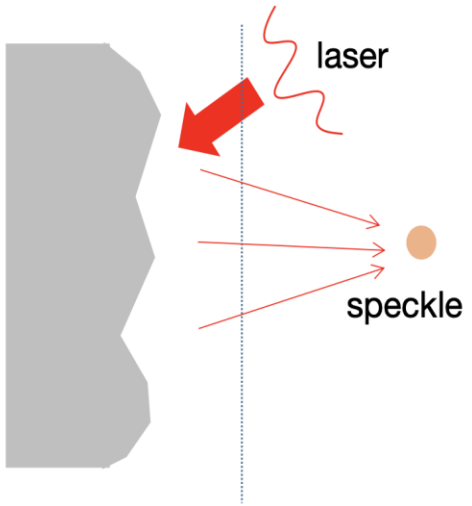
研究实施过程

散斑图像基本原理

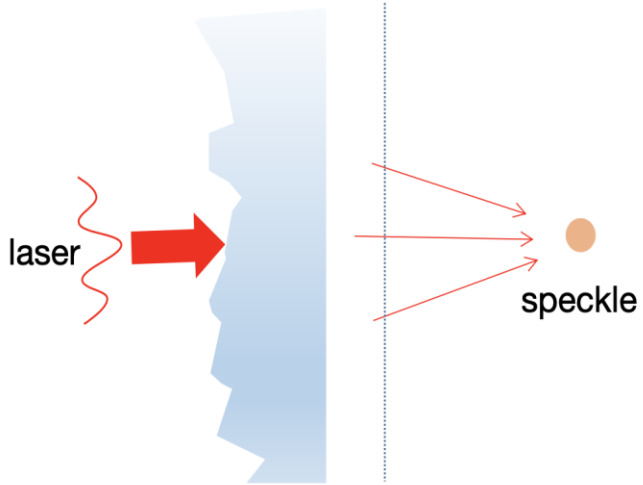
- 散斑图像成因
- 散斑图像的透射矩阵描述
- 传统散斑图像恢复方法

散斑图像成因

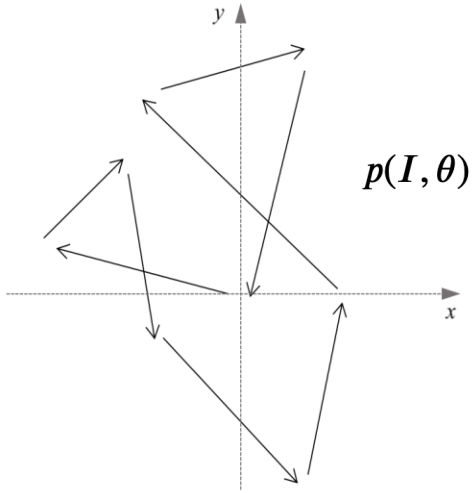
- 散斑形成有两个要素:
  (i) 光学粗糙的散射表面或者光学粗糙的（折射率不均匀的）透射介质的性质;
  (ii) 照明光场的相干性质。
- 无规则分布的光学粗糙界面或者无规则的介质的透射面元所散射的子光波进行叠加从而使得光场具有随机的空间光强分布，从而显示出精细的颗粒状或者斑纹图案。



(a)



(b)



(c)

$$p(I,\theta)=\begin{cases}\frac{1}{4\pi\sigma^2}e^{-\frac{I}{2\sigma^2}}, I\geqslant 0, -\pi\leqslant\theta<\pi,\\ 0, \quad \text{其他情况.}\end{cases}$$

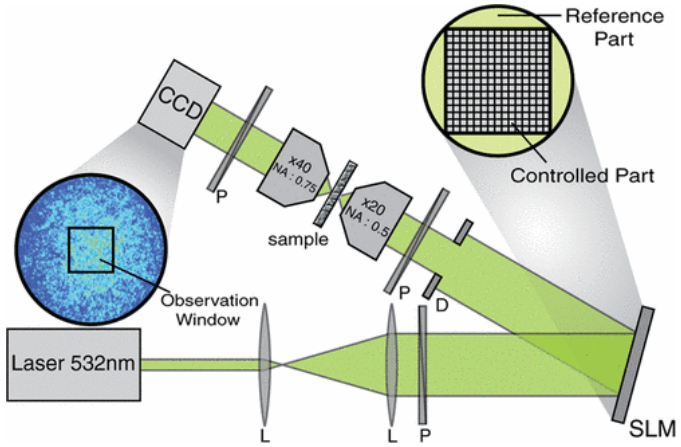
$$p(A)=\frac{A}{\sigma^2}e^{-\frac{A^2}{2\sigma^2}}, A\geqslant 0.$$

散斑图像的透射矩阵描述

- 将入射光场和出射光场离散化，利用线性近似，可以得到出射光场和入射光场之间的关系

$$A_k^{in} = A_k^{in}(\vec{r}_k^{in}) = \int_{S_k^{in}} d\vec{r} A^{in}(\vec{r}) \qquad A_l^{out} = A_l^{out}(\vec{r}_l^{out}) = \int_{S_l^{out}} d\vec{r} A^{out}(\vec{r}). \qquad A_l^{out} = \sum_{k=1}^{N} T_{lk} A_k^{in}$$

- 测量透射矩阵：四步相移干涉法；三步相移干涉法和五步相移干涉法



$$I_m^{\alpha} = \left|A_m^{\text{out}}\right|^2 = \left|s_m + \sum_n e^{i\alpha} T_{mn} A_n^{\text{in}}\right|^2$$

$$= \left|s_m\right|^2 + \left|\sum_n e^{i\alpha} T_{mn} A_n^{\text{in}}\right|^2 + 2\,\text{Re}\left(e^{i\alpha} \bar{s}_m \sum_n T_{mn} A_n^{\text{in}}\right)$$

$$I_m^0 = \left|s_m\right|^2 + \left|\sum_{n=1}^{N} T_{mn} A_n^{in}\right|^2 + 2\,\text{Re}\left(\bar{s}_m \sum_{n=1}^{N} T_{mn} A_n^{in}\right)$$

$$I_m^{\pi/2} = \left|s_m\right|^2 + \left|\sum_{n=1}^{N} T_{mn} A_n^{in}\right|^2 + 2\,\text{Re}\left(i\bar{s}_m \sum_{n=1}^{N} T_{mn} A_n^{in}\right)$$

$$I_m^{\pi} = \left|s_m\right|^2 + \left|\sum_{n=1}^{N} T_{mn} A_n^{in}\right|^2 - 2\,\text{Re}\left(\bar{s}_m \sum_{n=1}^{N} T_{mn} A_n^{in}\right)$$

$$I_m^{3\pi/2} = \left|s_m\right|^2 + \left|\sum_{n=1}^{N} T_{mn} A_n^{in}\right|^2 - 2\,\text{Re}\left(i\bar{s}_m \sum_{n=1}^{N} T_{mn} A_n^{in}\right)$$

$$\frac{I_m^0 - I_m^{\pi}}{4} + i\frac{I_m^{3\pi/2} - I_m^{\pi/2}}{4}$$

$$= \frac{2\,\text{Re}\left(2\bar{s}_m \sum_{n=1}^{N} T_{mn} A_n^{in}\right)}{4} + i\frac{2\,\text{Re}\left(-i2\bar{s}_m \sum_{n=1}^{N} T_{mn} A_n^{in}\right)}{4}$$

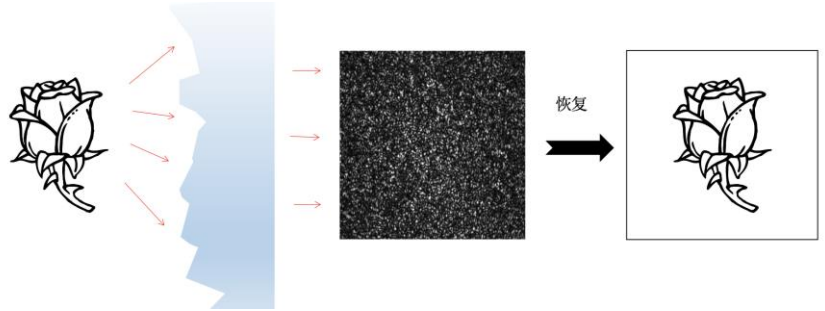
$$= \text{Re}\left(\bar{s}_m \sum_{n=1}^{N} T_{mn} A_n^{in}\right) + i\,\text{Im}\left(\bar{s}_m \sum_{n=1}^{N} T_{mn} A_n^{in}\right)$$

$$= \bar{s}_m \sum_{n=1}^{N} T_{mn} A_n^{in}.$$

$$T_{obs} = S_{ref} T$$

传统散斑图像恢复方法（基于透射矩阵测量）

- 散斑图像恢复问题



$$A_{obs}^{out} = T A^{in} + A_e$$

- 从测量的透射矩阵出发恢复散斑图像

1. 直接求逆方法（Moore-Penrose逆）

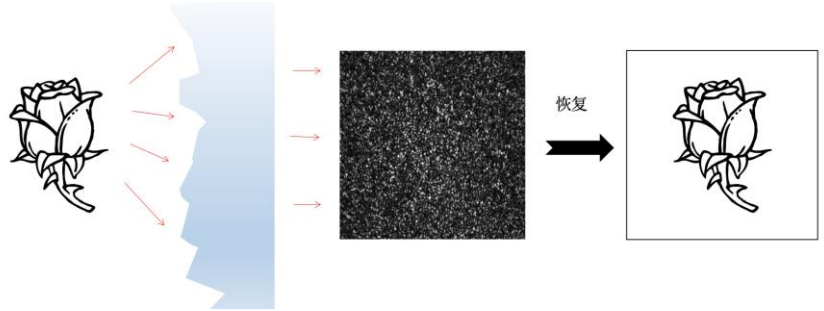
$$T_{obs}^{-1} = (T_{obs}^{\dagger} T_{obs})^{-1} T_{obs}^{\dagger} \qquad \hat{A} = T_{obs}^{-1} A_{obs}^{out} = (T + T_e)^{-1} (T A^{in} + A_e)$$

2. 光学相位共轭方法 $\hat{A} = T_{obs}^{\dagger} A_{obs}^{out} = T_{obs}^{\dagger} (T A^{in} + A_e)$

3. Tikhnov 正则化算法 $\hat{A} = W A_{obs}^{out} \qquad W = (T_{obs}^{\dagger} T_{obs} + \mu I)^{-1} T_{obs}^{\dagger}$

传统散斑图像恢复方法（基于场强的方法）

- 散斑图像恢复问题



- 基于场强的方法（散斑关联矩阵）

$$\mathbf{Z}_{pq} = \frac{1}{\Sigma_p \Sigma_q}\left[\left\langle t_p^* t_q (E^{out})^* E^{out}\right\rangle_r - \left\langle t_p^* t_q\right\rangle_r \left\langle (E^{out})^* E^{out}\right\rangle_r\right]$$

直接求散斑关联矩阵的本征值最大的本征向量，就是待求的光场。

传统方法的缺陷：

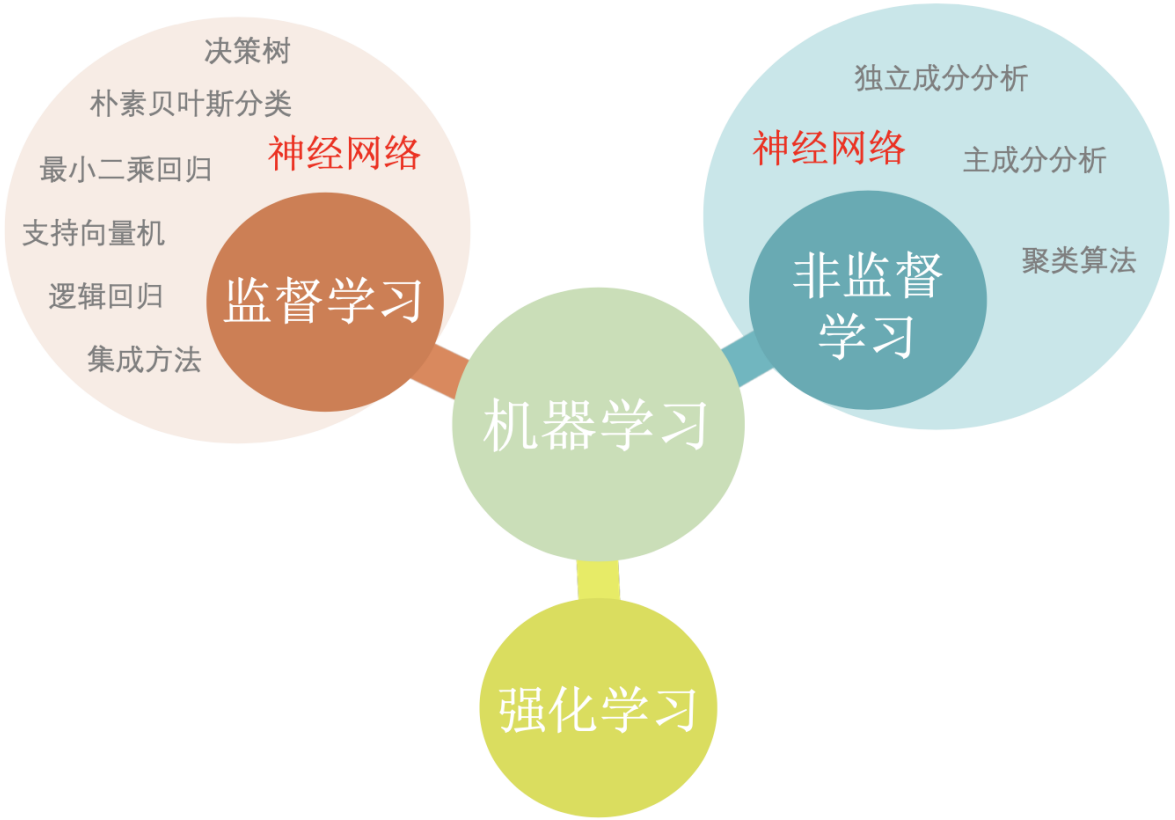
（a）需要测量透射矩阵，这涉及大量的测量
（b）需要对不同介质分别测量透射矩阵，恢复过程是一对一的

神经网络图像处理

- *机器学习与神经网络*
- *卷积神经网络*
- *基于卷积神经网络的图像处理*

中国科学技术大学
University of Science and Technology of China

机器学习与神经网络



中国科学技术大学
University of Science and Technology of China

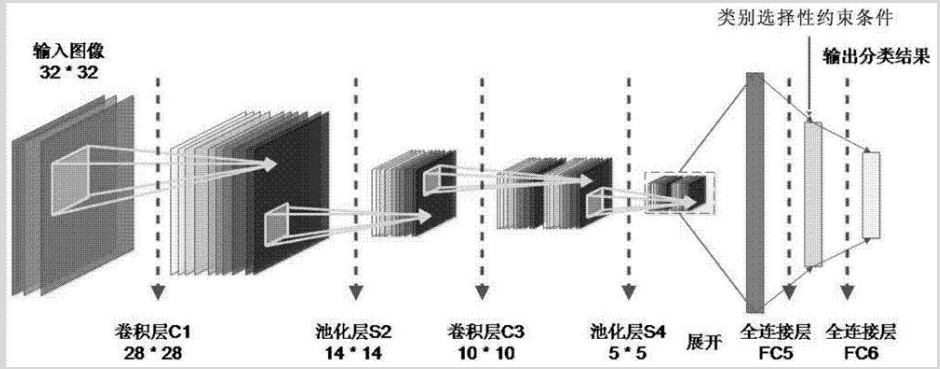
神经网络

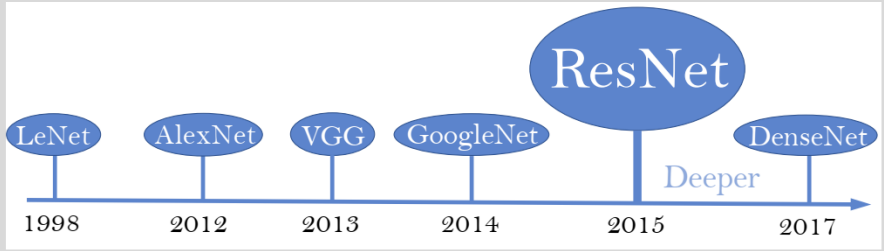
• 前馈神经网络 (feedforward neural network)，它包括各种感知机的变形，误差逆传播 (error backpropagation) 神经网络，径向基函数 (radial basis function) 神经网络，以及我们将着重讨论的卷积神经网络等。

• 统计回流神经网络 (stochastic recurrent neural network)，它包括 Hopflied 神 经网络 (Hopfield neural network)，各种 Boltzmann 机 (Boltzmann machine) 的变形等等。

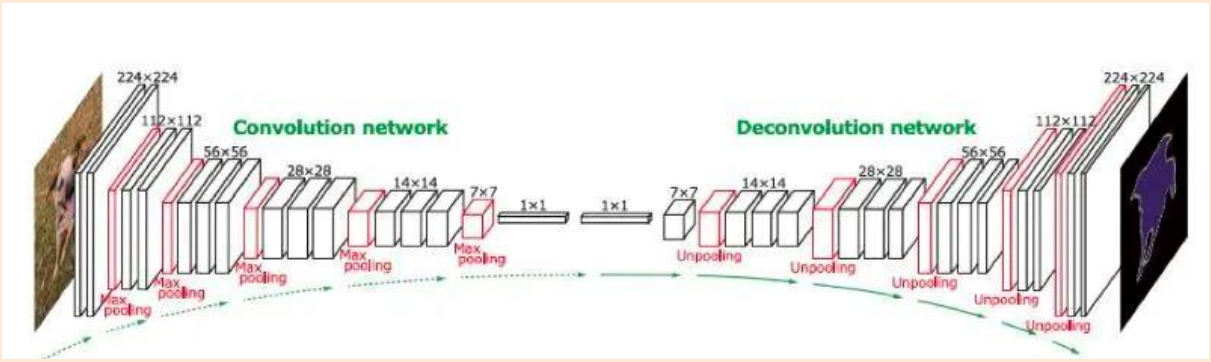
卷积神经网络

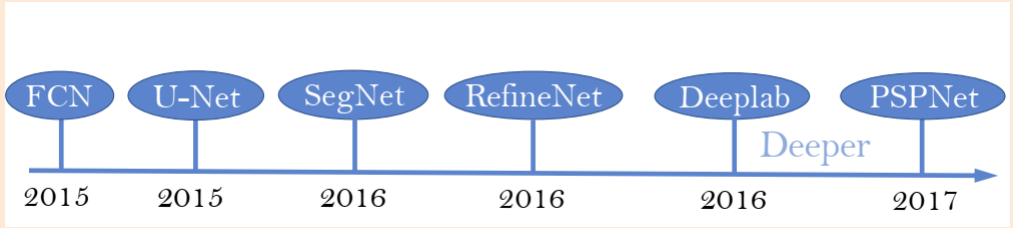
分类器





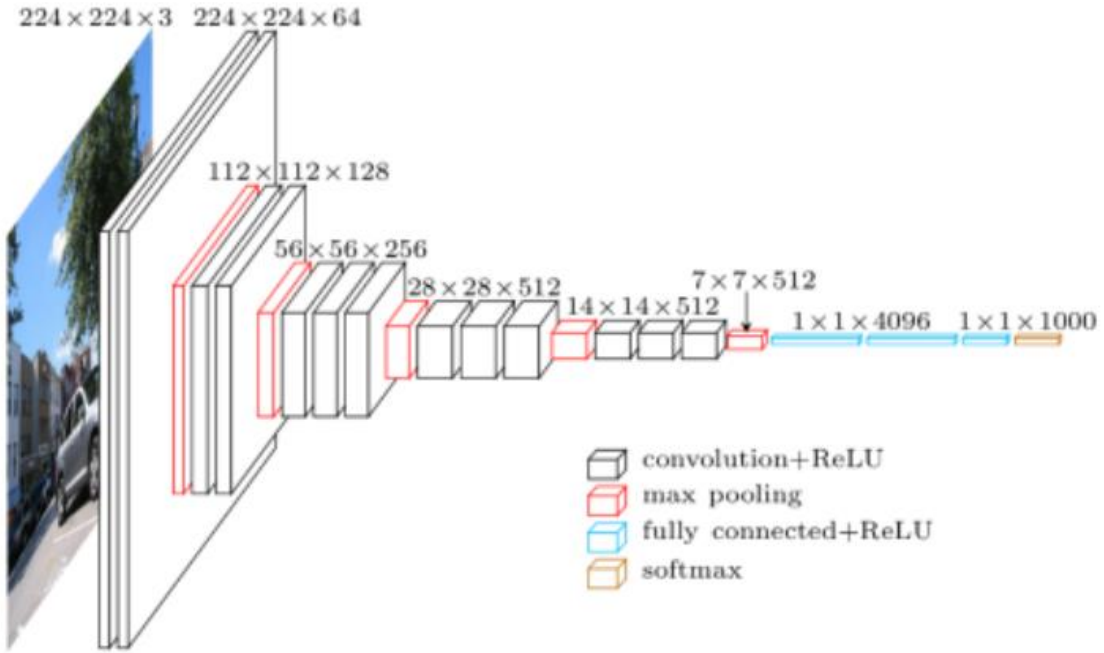
语义分割





基于卷积神经网络的图像处理

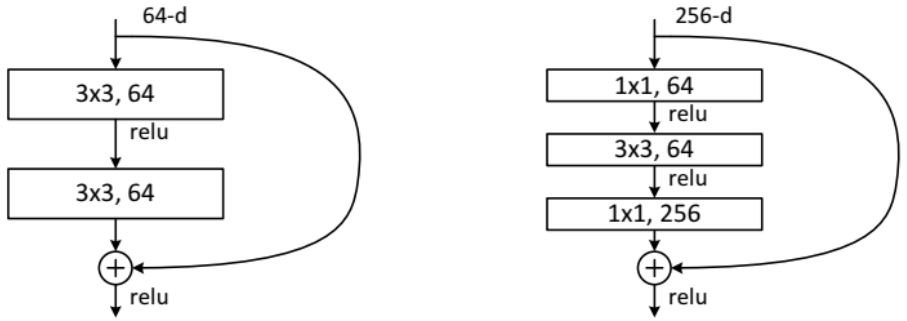
VGG架构



VGG是在 AlexNet 上进行改进的，采用连续的几个 3 × 3 的卷积核代替原来 AlexNet 中的较大的 11 × 11、 7 × 7、 5 × 5 的卷积核，增加网络深度并减少参数个数来学习完成更复杂的任务。其中权重数目最多的地方就在第一个全连接层中，有许多公开的预训练权重包含的就是这些全连接层的权重。

基于卷积神经网络的图像处理

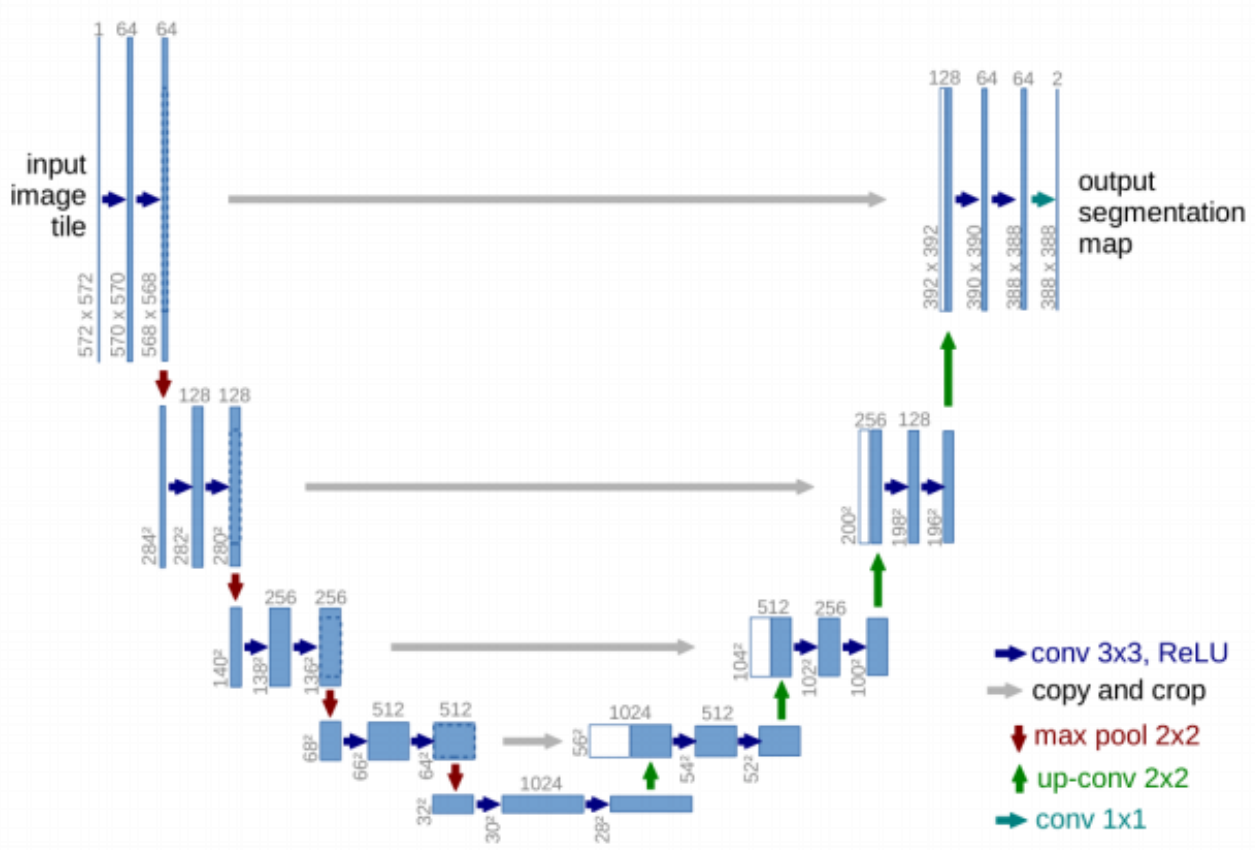
ResNet架构



图左是在 ResNet34 中的基本构成要素。图右是在 ResNet50/101/152 中的“瓶颈”基本构成要素。

基于卷积神经网络的图像处理

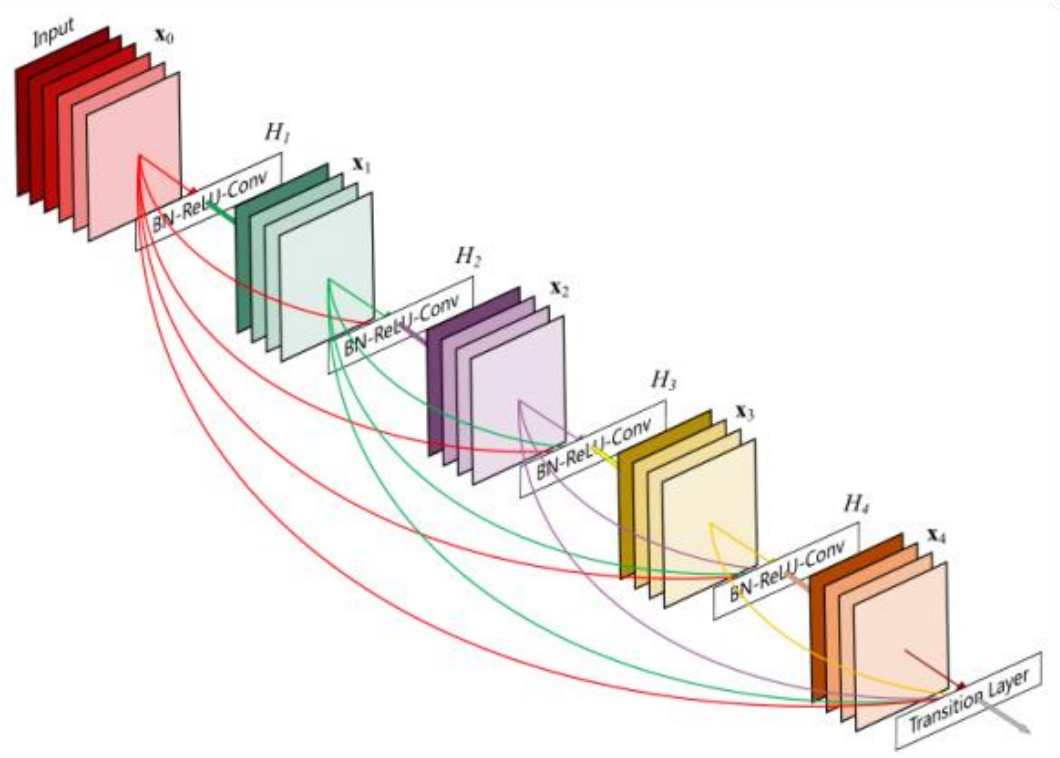
Unet架构



Unet的网络架构，蓝色方框代表一个多通道特征图，通道数在盒子顶部，输入尺寸在盒子底部，白色方框代表复制拼接的特征图，蓝色的箭头代表做 3 × 3 卷积以后再进行线性整流，灰色箭头代表复制收缩路径每一步的特征层后直接拼接到膨胀路径每一步对应的输入中 (左边的虚线框就代表复制操作)，红色箭头代表了 2 × 2 的最大下采样，绿色箭头代表了 2 × 2 的反卷积操作，浅蓝色的箭头代表了 1 × 1 的卷积操作，该网络所能容许的最低分辨率为 32 × 32。

基于卷积神经网络的图像处理

DenseNet架构



五层的密集块，其增长率为 4，每层的输入都是前面所有层的输出。

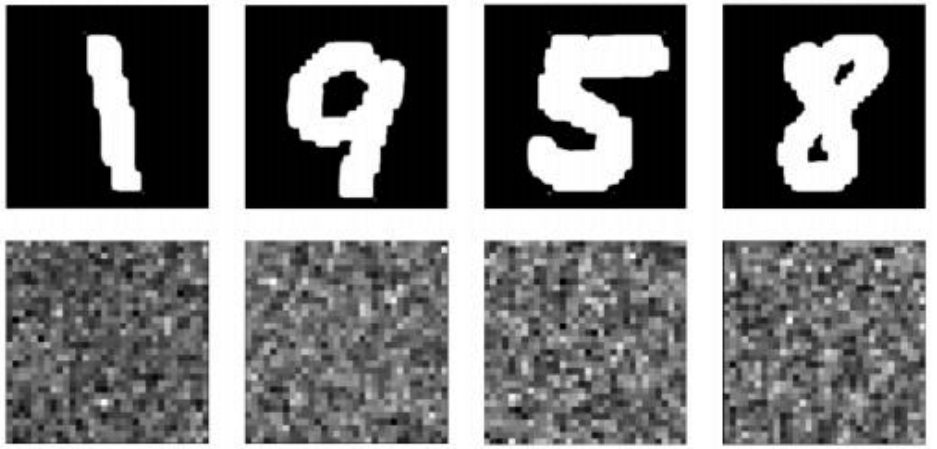
神经网络散斑图像处理

- 散斑图像生成的数值结果
- 神经网络架构以及训练
- 散斑图像恢复结果讨论
- 后续工作展望

散斑图像生成的数值结果

- 利用透射矩阵生成散斑图像 $A_l^{out} = \sum_{k=1}^{N} T_{lk} A_k^{in}$

  Note：这里没考虑噪声，$A_{obs}^{out} = T A^{in} + A_e$

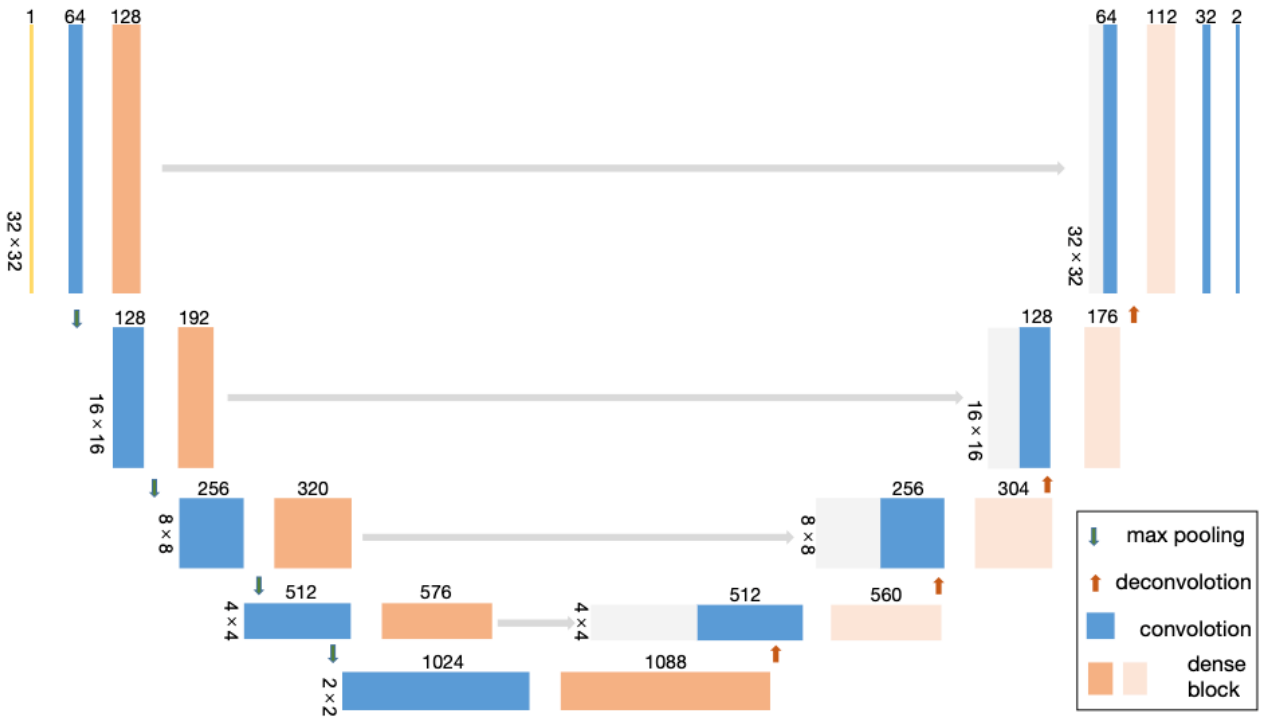


- 手写数字识别数据集，MNIST
  http://yann.lecun.com/exdb/mnist/
- 用Matlab进行数据生成，生成散斑图片的核心代码如下：

```
TM = raylrnd(sqrt(1/2/M),M,N).*exp(1i * random('unif',-pi,+pi-eps('double'),M,N)); % transmission matrix generation.
TM_prac = awgn(TM,SNR,'measured','linear'); % add noise upon preset SNR.
disp(['TM generate end! ... Elapsed time: ', num2str(toc),'s']);
Y = TM * X; % diffused field y generation.
Y_prac = awgn(Y,SNR,'measured','linear'); % add noise upon preset SNR.
Y_prac2d = reshape(Y_prac,[sqrt(M),sqrt(M)]); % 2D reshape. #for visualization.
Iyy = abs(Y_prac).^2; % measured intensity speckle single shot.
```

- 过程中遇到的问题
- 结果

神经网络架构以及训练



神经网络架构

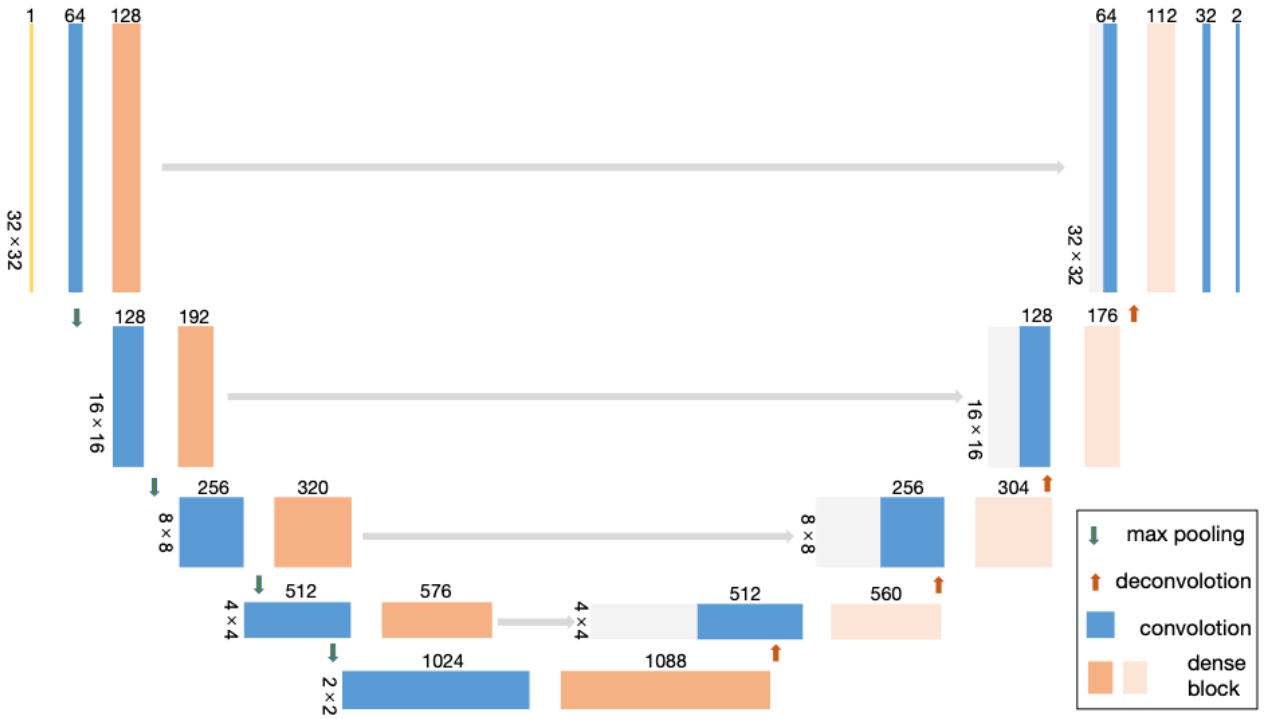
架构依据：

- 全连接浅层神经网络
- 卷积神经网络
- Unet
- DenseNet
- IDiffNet
- Unet+DenseNet

优势：

同时结合了Unet和DenseNet的优点，能够反复使用提取出的特征层，而且很大程度上降低了网络的参数数目，使得网络层数能够加深。

神经网络架构以及训练



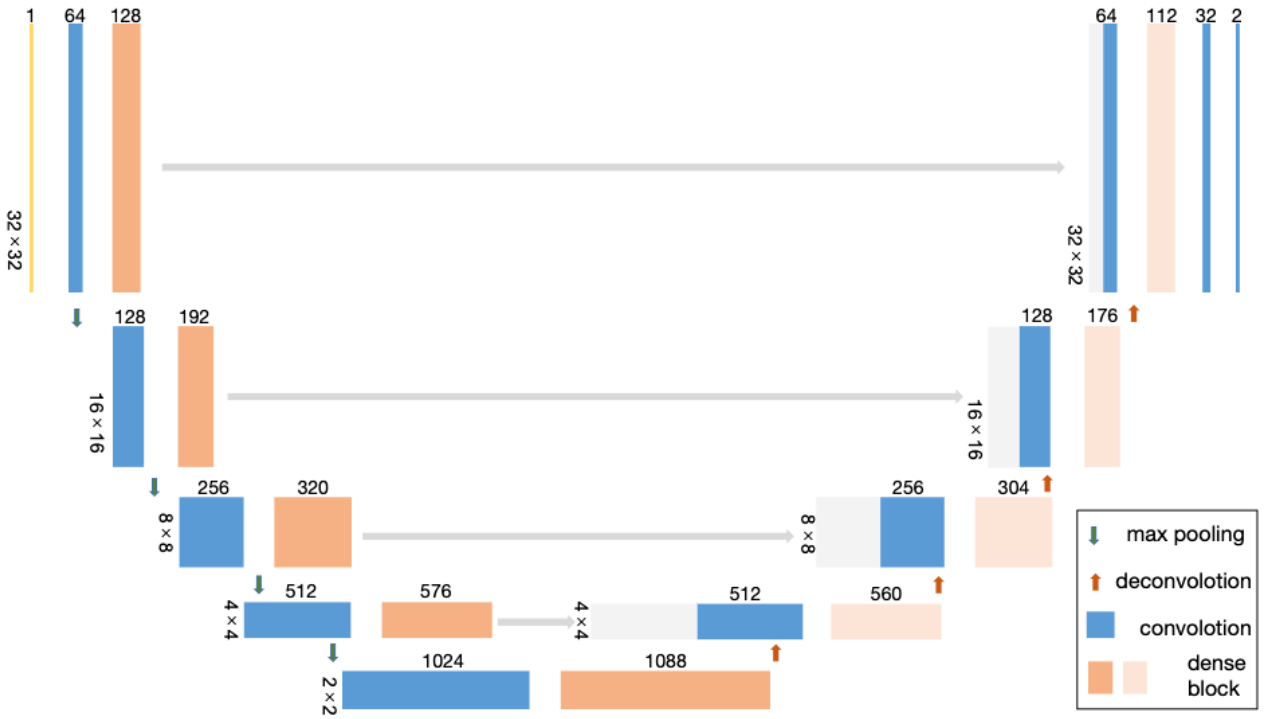
核心代码

```
inputs = Input((64, 64, 1))
    print("inputs shape:", inputs.shape)

    conv1 = Conv2D(64, 3, activation='relu', padding='same',
kernel_initializer='he_normal')(inputs)
    print("conv1 shape:", conv1.shape)
    db1 = denseblock(x=conv1, concat_axis=3, nb_layers=4, growth_rate=16,
dropout_rate=0.5)
    print("db1 shape:", db1.shape)
    pool1 = MaxPooling2D(pool_size=(2, 2))(db1)
    print("pool1 shape:", pool1.shape)
      ⋮
    conv5 = Conv2D(1024, 3, activation='relu', padding='same',
kernel_initializer='he_normal')(pool4)
    print("conv5 shape:", conv5.shape)
    db5 = denseblock(x=conv5, concat_axis=3, nb_layers=4, growth_rate=16,
dropout_rate=0.5)
    print("db5 shape:", db5.shape)
    up5 = Conv2D(512, 2, activation='relu', padding='same',
kernel_initializer='he_normal')(
        UpSampling2D(size=(2, 2))(db5))
    print("up5 shape:", up5.shape)
    merge5 = Concatenate(axis=3)([db4, up5])
    print("merge5 shape:", merge5.shape)

    conv6 = Conv2D(512, 3, activation='relu', padding='same',
kernel_initializer='he_normal')(merge5)
    print("conv6 shape:", conv6.shape)
    db6 = denseblock(x=conv6, concat_axis=3, nb_layers=3, growth_rate=16,
dropout_rate=0.5)
    print("db5 shape:", db6.shape)
    up6 = Conv2D(256, 2, activation='relu', padding='same',
kernel_initializer='he_normal')(
        UpSampling2D(size=(2, 2))(db6))
    print("up6 shape:", up6.shape)
    merge6 = Concatenate(axis=3)([db3, up6])
    print("merge6 shape:", merge6.shape)
      ⋮
    model = Model(inputs=inputs, outputs=conv11)
```

神经网络架构以及训练



中国科学技术大学
University of Science and Technology of China

训练过程

数据预处理：

将matlab生成的mat数据文件使用numpy写成可以处理的矩阵，使用Z归一化更新训练数据来加速训练。

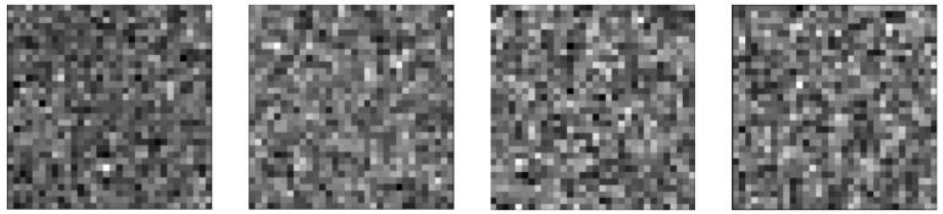
训练参数细节：

- 优化器：adam
- 更新步长：1e-4
- 损失函数：categorical_crossentropy
- 指标：accuracy
- 模型保存方式：save_weights_only
- Epoch：300
- batch_size：1-35为20；36-最优为10

散斑图像恢复结果讨论

原始图片

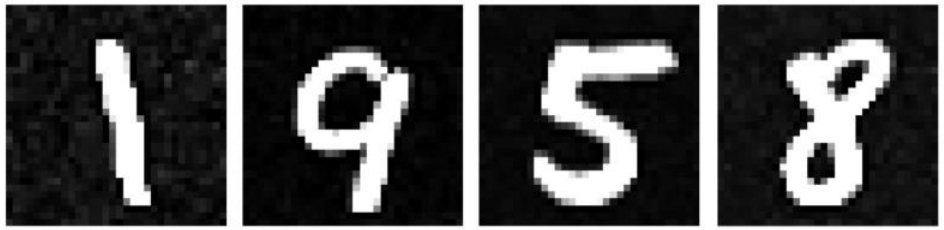
散斑图片



神经网络恢复图片

场强方法恢复图片

透射矩阵恢复图片



训练过程

神经网络方法还原散斑图片：

- 还原效果很好，细节处理得也很好，准确度可达到0.6321
- 不需要测量透射矩阵
- 可以针对多种不同介质同时训练，从而对不同介质的图片同时恢复

传统方法还原散斑图片：

- 引入噪声后进行模拟
- 细节处理得不够好，仍然携带了许多散斑信息
- 针对不同介质必须做不同的设计和测量

后续工作展望

- 神经网络预处理问题：还没有人做神经网络预分类再还原的工作，这部分是可以继续做的。
- 实验方面：针对不同介质生成散斑图像数据库，利用空间光调制器生成用于训练的散斑图像数据库。
- 换用不同架构：可以尝试更改神经网络架构。
- 实时图像还原：由于经过训练的神经网络权重可以还原通过散射介质的其他点形成的散斑，可以用来做更加实用的实时图像还原。

总结

- 系统调研了散斑图像恢复的问题
- 系统调研了深度神经网络在计算机视觉中的应用
- 尝试使用深度神经网络对散斑图像进行处理并获得了较好的结果

研究实施过程

选题调研 — 1月-2月

1.调研散斑图像的各个方面的内容
2.调研了深度学习图像处理的相关架构

1月

选定题目《基于深度学习方法实现透过散射介质成像》，并调研的这个领域的研究现状

调研了散斑图像的物理成因、传统的散斑图像恢复方法：直接求逆方法；光学相位共轭方法，Tikhnov正则化算法，场强方法

2月

初步整理了传统散斑图像恢复的方法，对基于深度学习的光学图像处理问题进行了调研

调研了常见的深度学习图像处理的神经网络架构

数值研究 — 3月-5月

1.选定了结合了Unet和DenseNet的神经网络架构来实现散斑图像恢复问题
2.利用透射矩阵的方法生成了散斑图像训练数据集
3.训练神经网络并进行测试和反复调试
4.利用传统方法恢复散斑图像，并与神经网络结果比较

3月

利用透射矩阵的方法进行了散斑图像生成，并将生成的数据进行了预处理，以便进行神经网络训练

进行了散斑图像生成的数值实验，利用MNIST数据集，结合透射矩阵的方法进行了散斑图像生成实验

4月

换用多个不同架构的神经网络进行了实验，最终选定了结合Unet和DenseNet的架构，对神经网络进行了训练和测试

架构了结合Unet和DenseNet的卷积神经网络，并利用生成的散斑图像训练数据集对神经网络进行了训练，测试和优化

4月-5月

利用传统散斑图像恢复方法进行了数值试验，并将结果与神经网络方法进行了比较

进一步优化了神经网络，并用传统的散斑图像恢复方法进行了数值实验，与神经网络的结果进行比较

论文撰写 — 5月

对神经网络进行了进一步的优化和调试，并撰写了毕业论文

End

致谢

导师王自强老师，李银妹老师
伍新明老师，徐云老师
实验室的各位师兄师姐

END

THANK YOU