

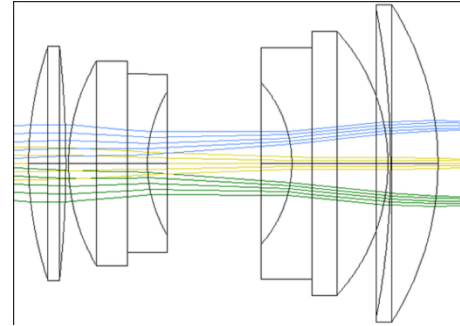
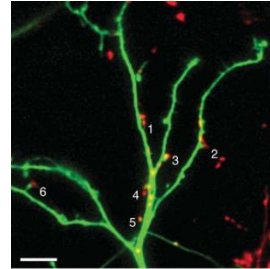
# Lecture 16: Coherent physical layers and general guidelines

Machine Learning and Imaging

BME 548L  
Roarke Horstmeyer

# Summary of two models for image formation

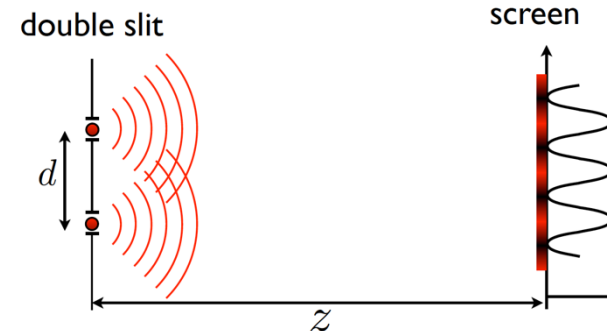
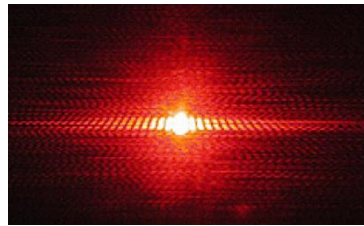
- Interpretation #1: Radiation (*Incoherent*)
- Model: Rays



- Real, non-negative
- Models absorption and brightness

$$I_{\text{tot}} = I_1 + I_2$$

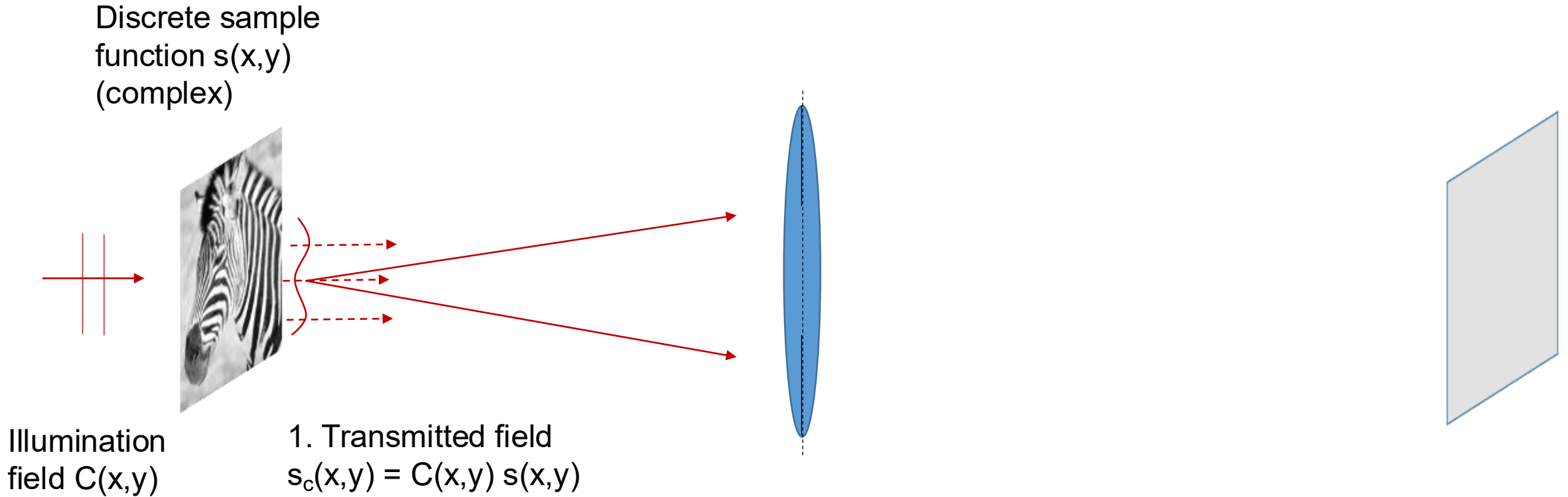
- Interpretation #2: Electromagnetic wave (*Coherent*)
- Model: Waves



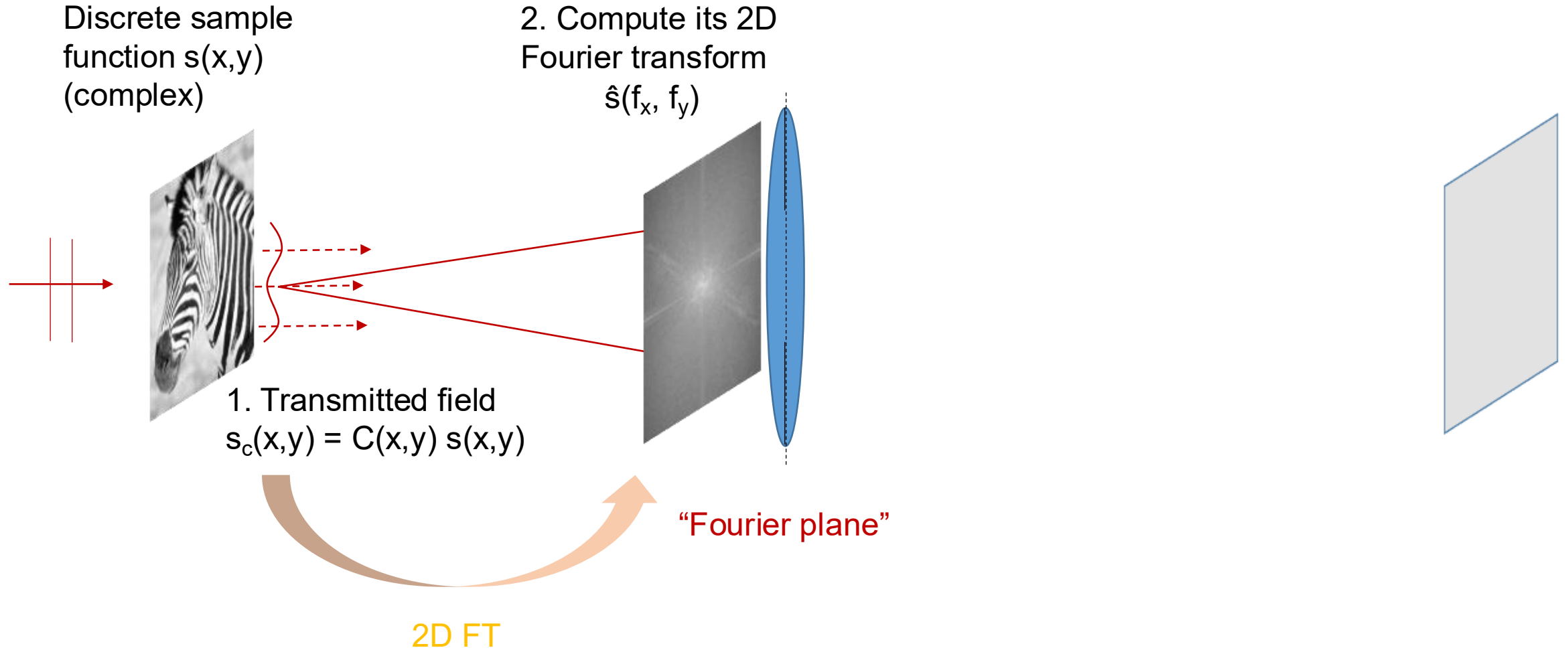
- Complex field
- Models Interference

$$E_{\text{tot}} = E_1 + E_2$$

# Model of image formation for wave optics (coherent light):

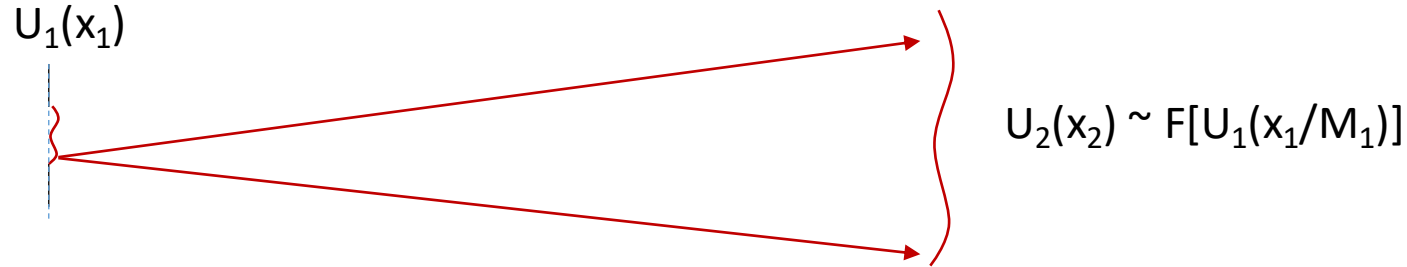


# Model of image formation for wave optics (coherent light):

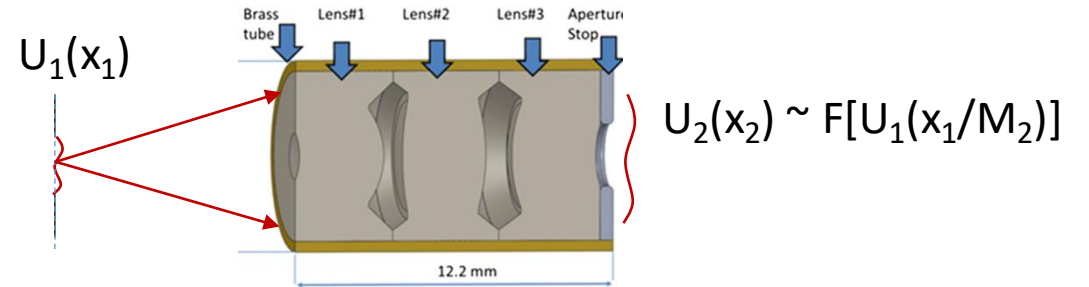


# General rules for applying the Fourier transform in optics

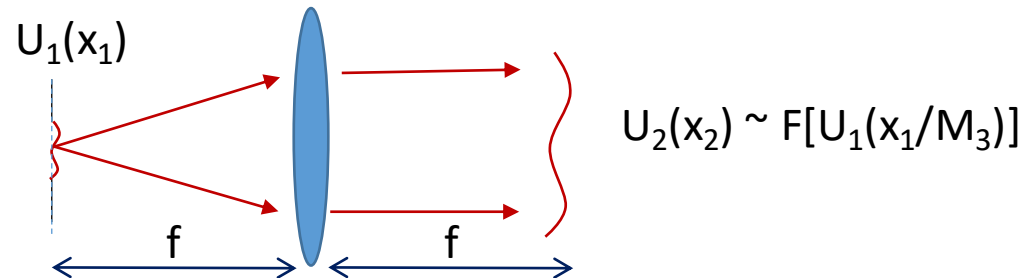
Situation 1: From an object to a plane “really far away”



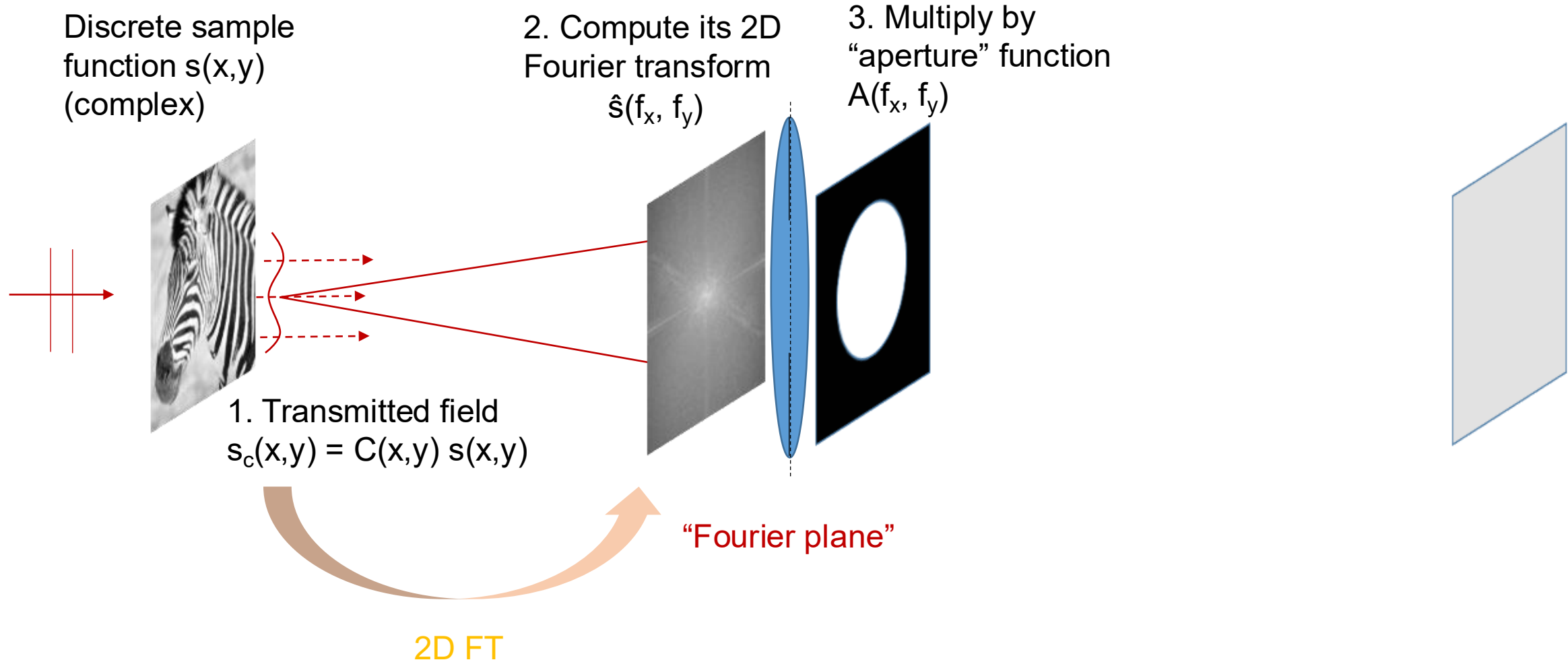
Situation 2: From an object to the back focal plane of the microscope objective lens



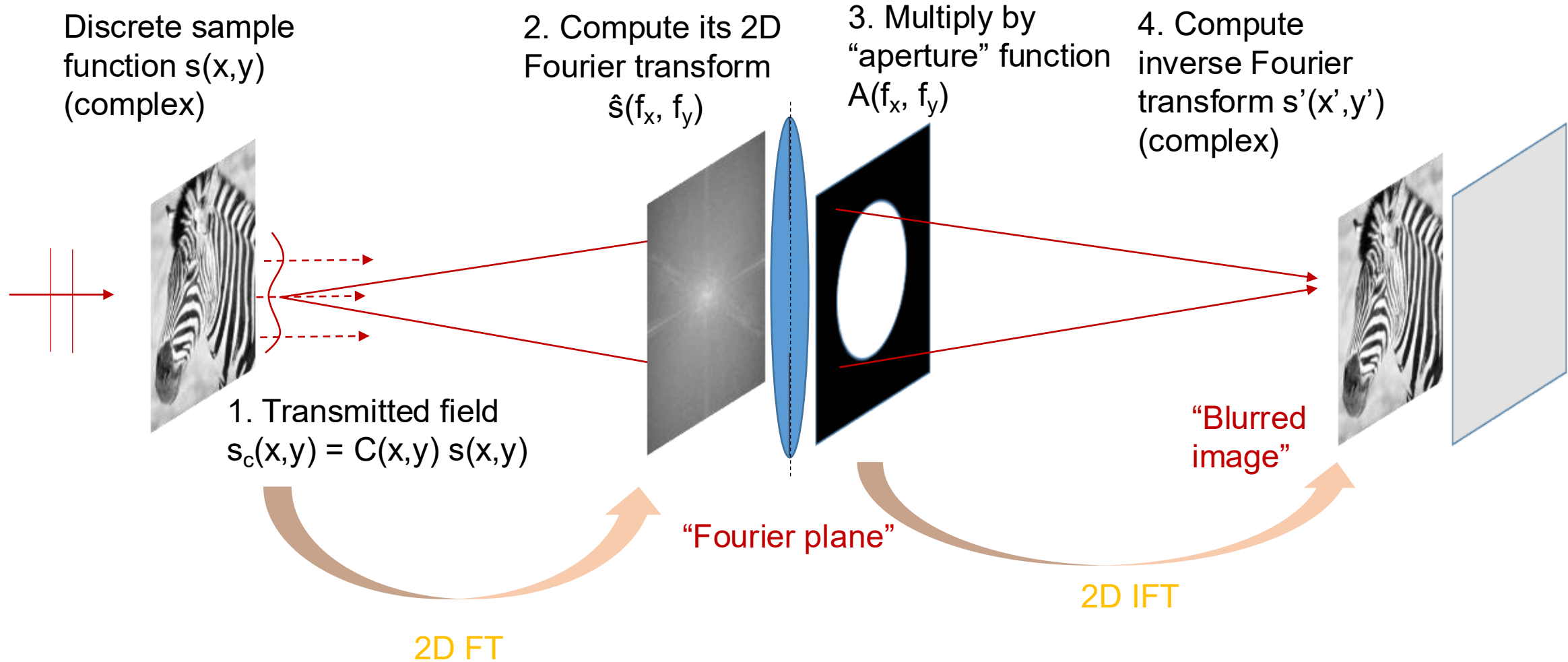
Situation 3: From an object to a plane 1 focal length away from a lens (1f-1f system)



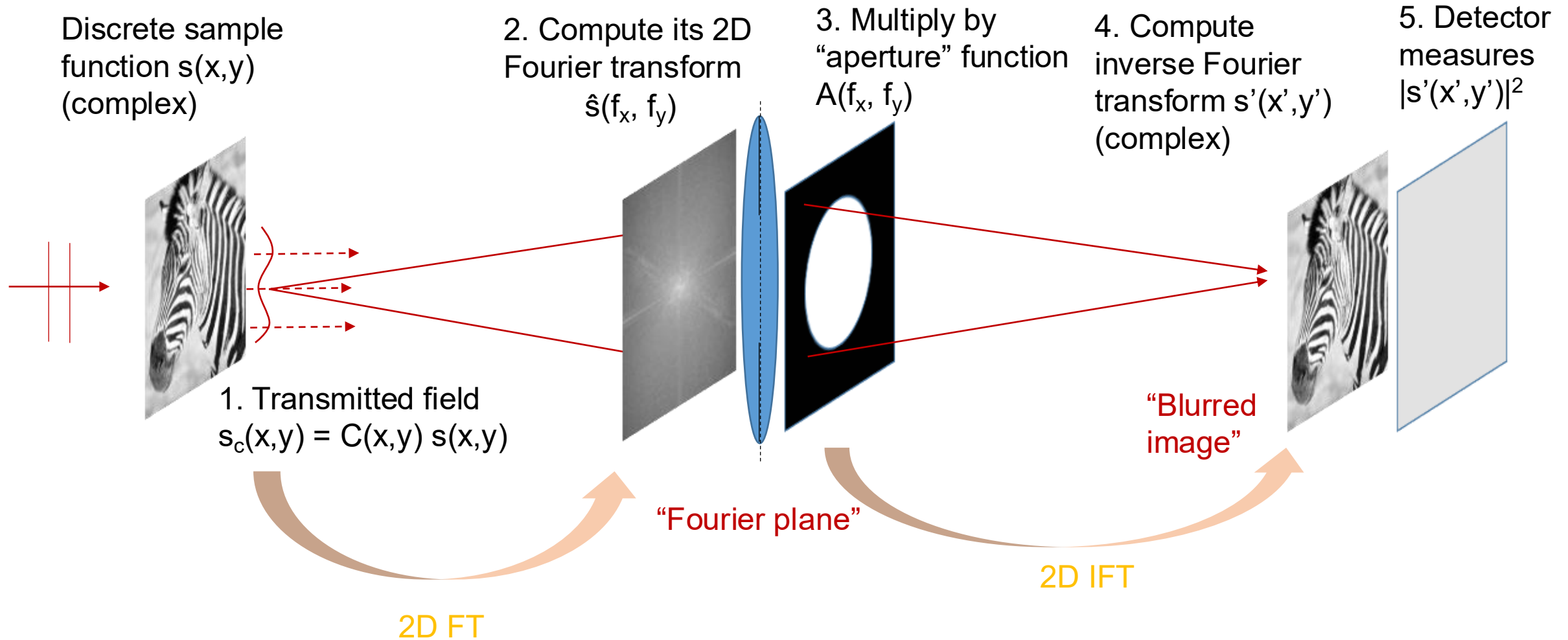
# Model of image formation for wave optics (coherent light):



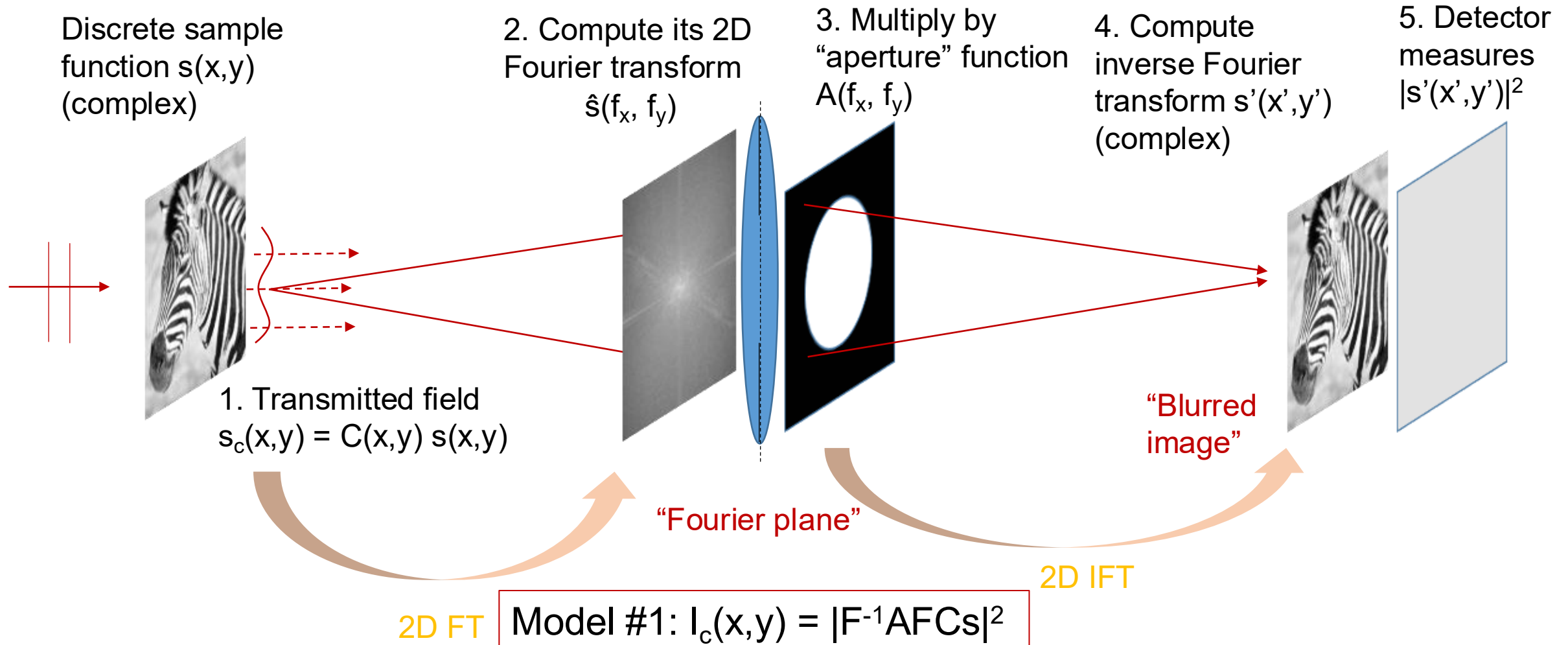
# Model of image formation for wave optics (coherent light):



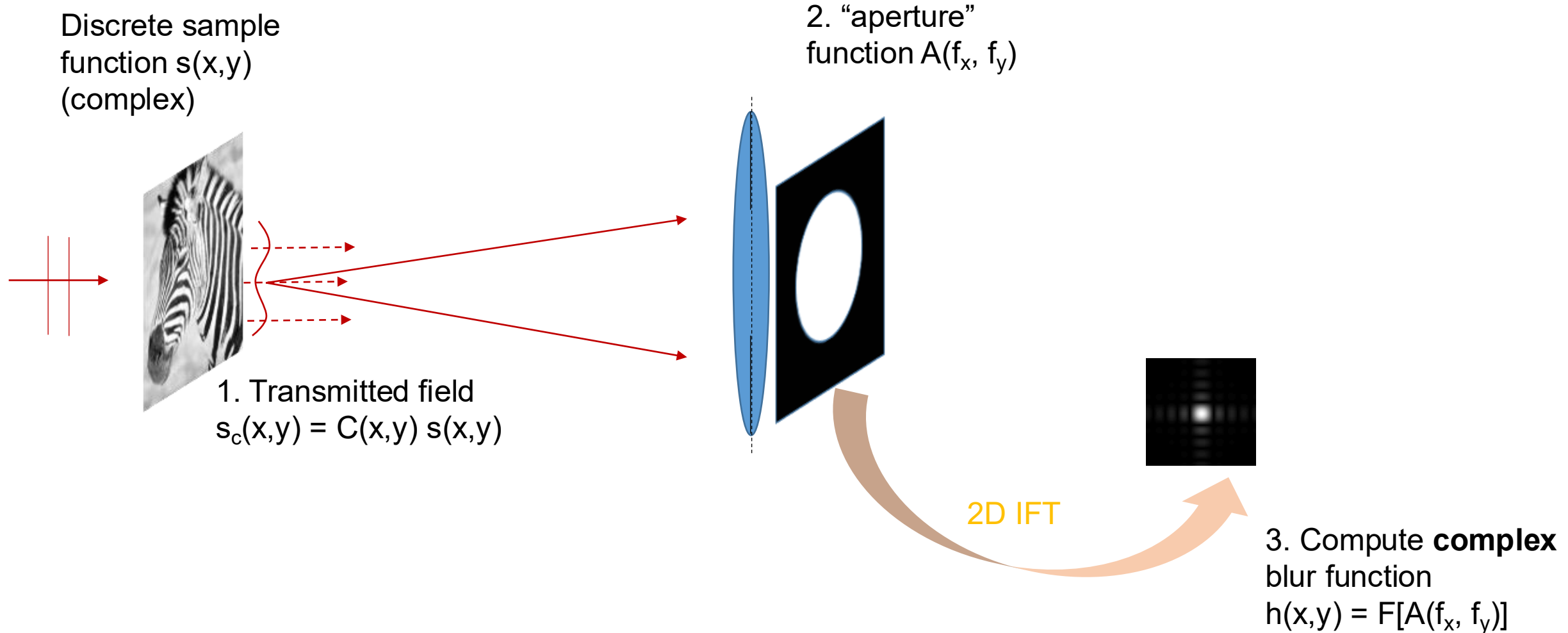
# Model of image formation for wave optics (coherent light):



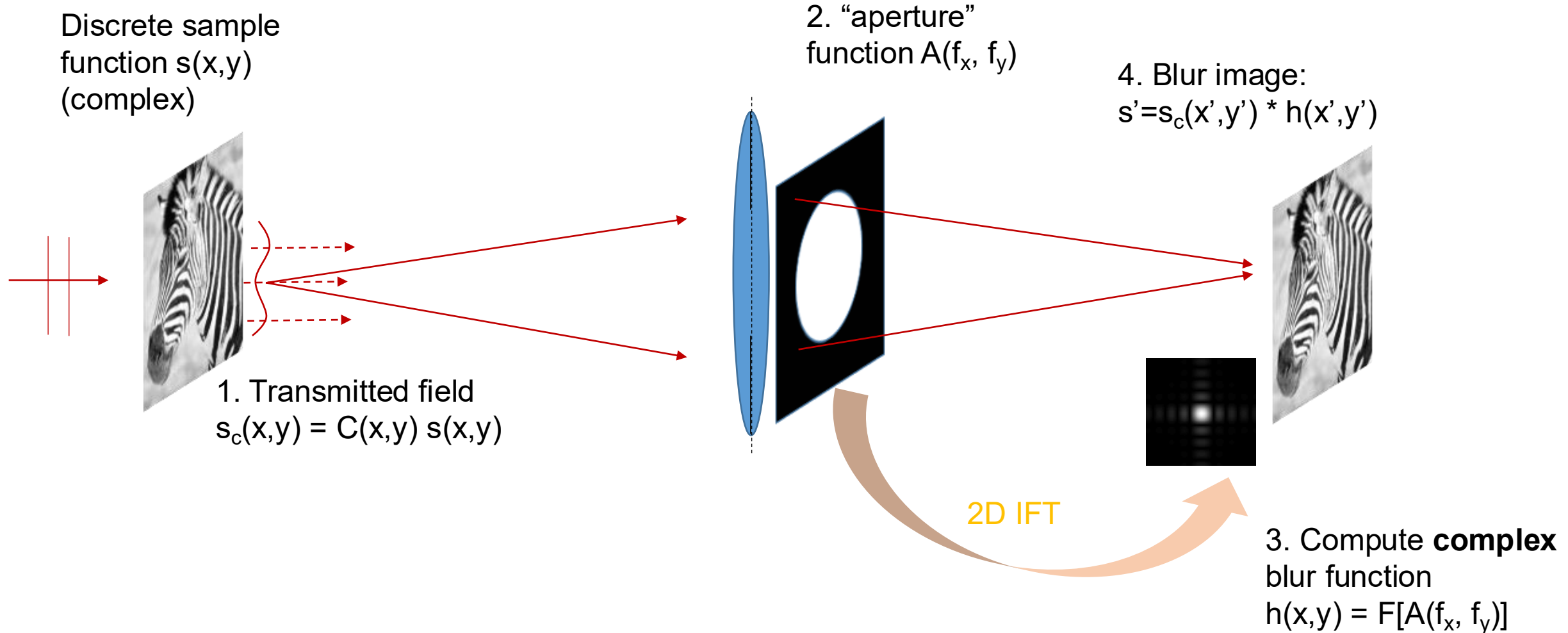
# Model of image formation for wave optics (coherent light):



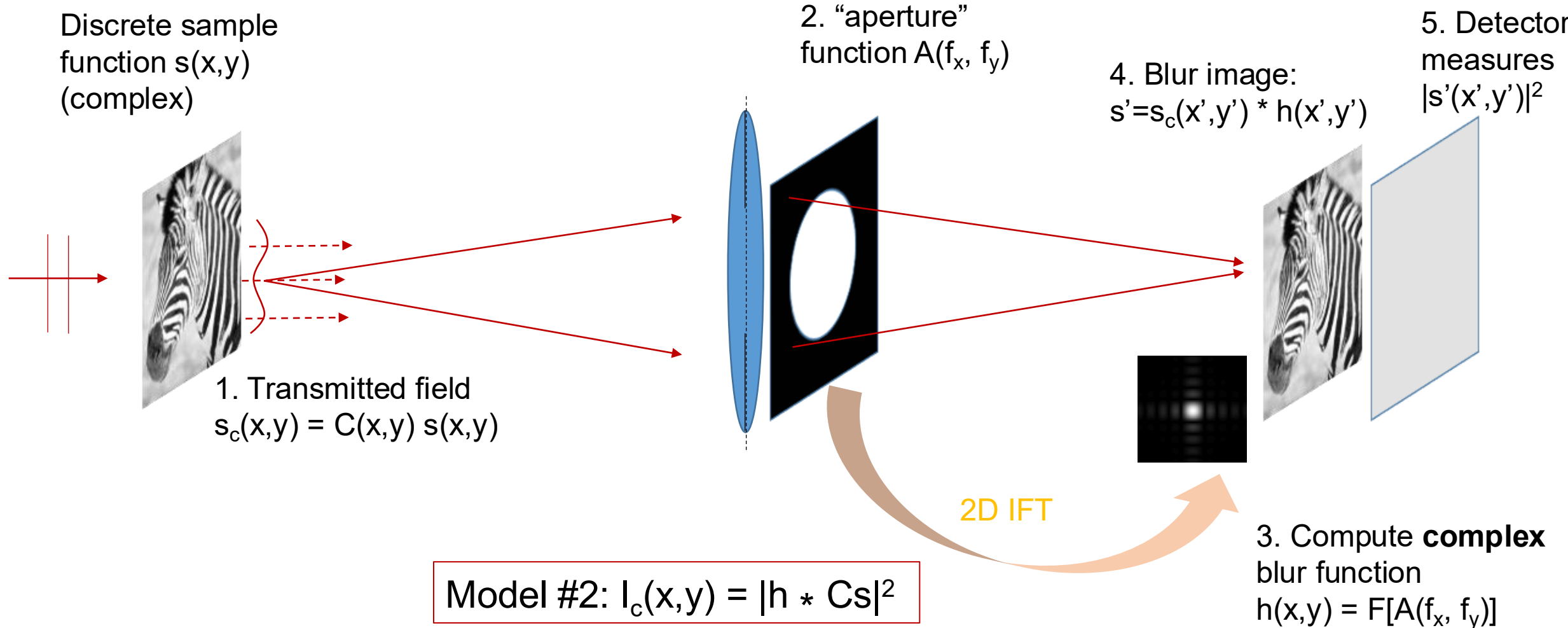
# Model of image formation for wave optics (coherent light):



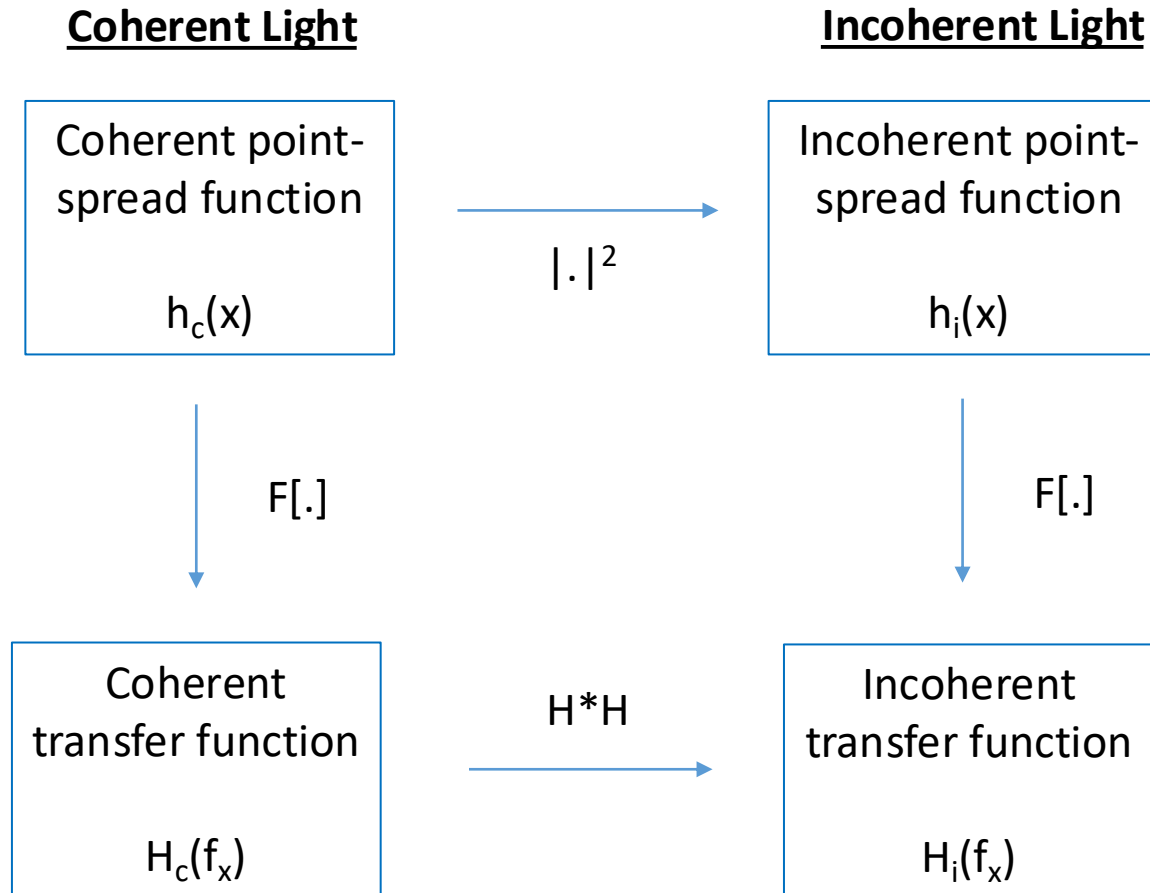
# Model of image formation for wave optics (coherent light):



# Model of image formation for wave optics (coherent light):



You typically go between 4 functions to describe one imaging system:

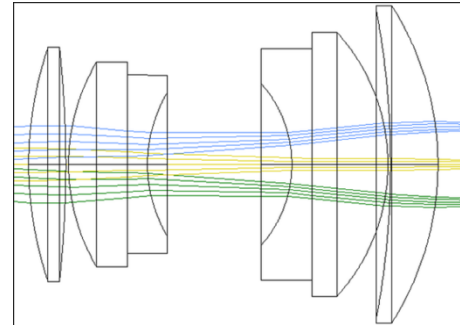
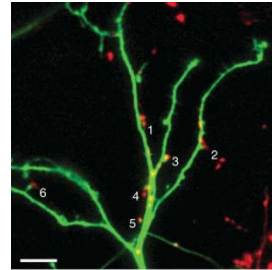


Incoherent PSF = Coherent PSF squared:

$$h_i(x) = |h_c(x)|^2$$

# Summary of two models for image formation

- Interpretation #1: Radiation (*Incoherent*)
- Model: Rays

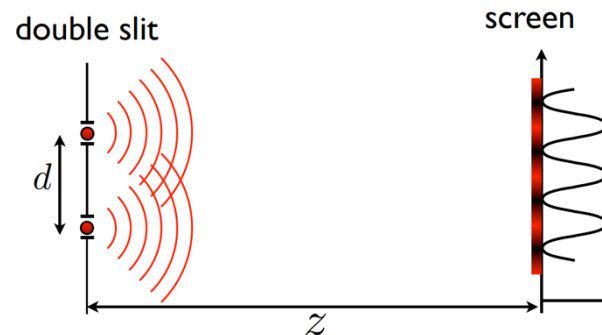
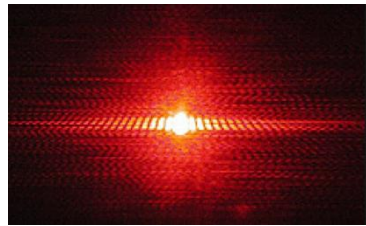


- Real, non-negative

$$I_s = h_i * B S_0$$

- Sample absorption **S**
- Illumination brightness **B**
- Blur in **h<sub>i</sub>** (real, positive)

- Interpretation #2: Electromagnetic wave (*Coherent*)
- Model: Waves



- Complex-valued

$$I_c = |h_c * C S_c|^2$$

- Sample abs./phase **S**
- Illumination wave **B**
- Blur in **h<sub>c</sub>** (complex)

## Example #1: Optimizing coherent illumination pattern for improved classification

Example future situation: Hacking has brought online banking to a halt. We now rely on a special form of physical check that is made of visibly transparent plastic. To write the amount in, you press down with a pen-like instrument, and then the check is read out by shining a particular pattern of laser light onto it, and then imaging it with a lens.

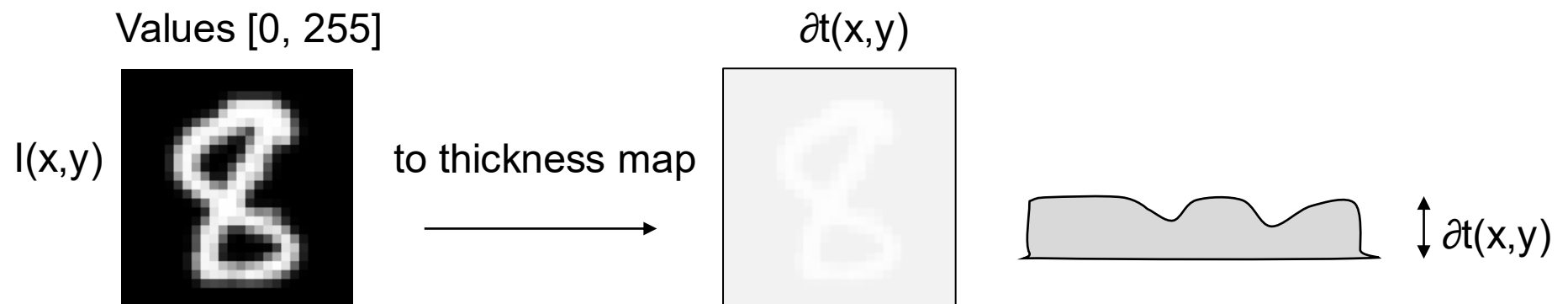
Question: What type of illumination should you use to maximize the classification accuracy of the numbers on the check?

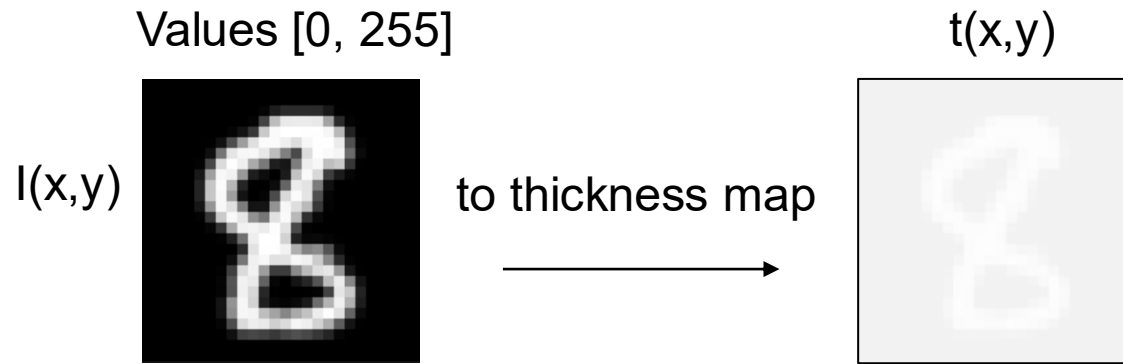
## Example #1: Optimizing coherent illumination pattern for improved classification

Example future situation: Hacking has brought online banking to a halt. We now rely on a special form of physical check that is made of visibly transparent plastic. To write the amount in, you press down with a pen-like instrument, and then the check is read out by shining a particular pattern of laser light onto it, and then imaging it with a lens.

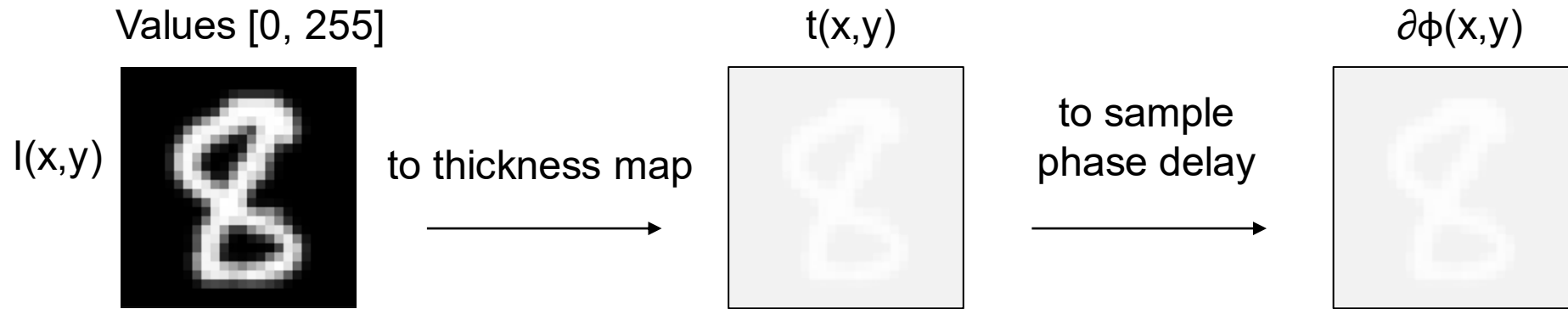
Question: What type of illumination should you use to maximize the classification accuracy of the numbers on the check?

### Step 1: Transform MNIST image data set into transparent plastic sheets with varying thickness

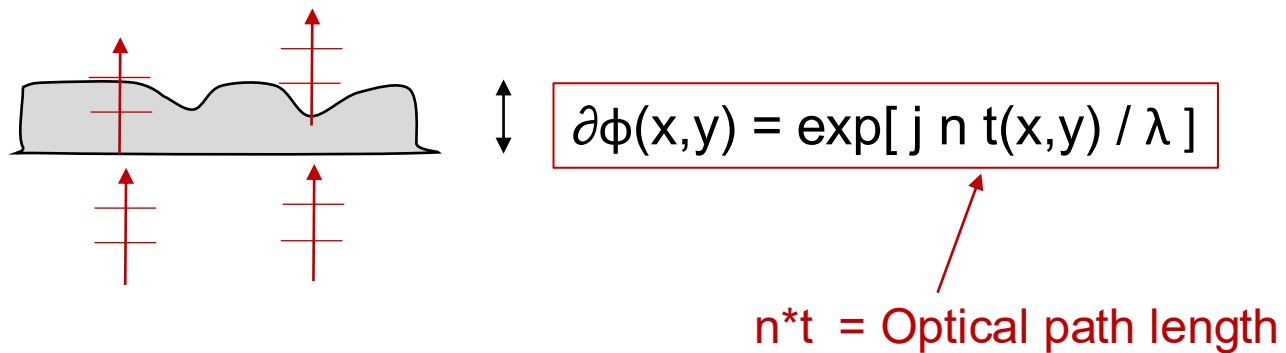




1. Normalize intensity map to 1
2. Define thickness map at some reasonable amount (100  $\mu\text{m}$  max change)

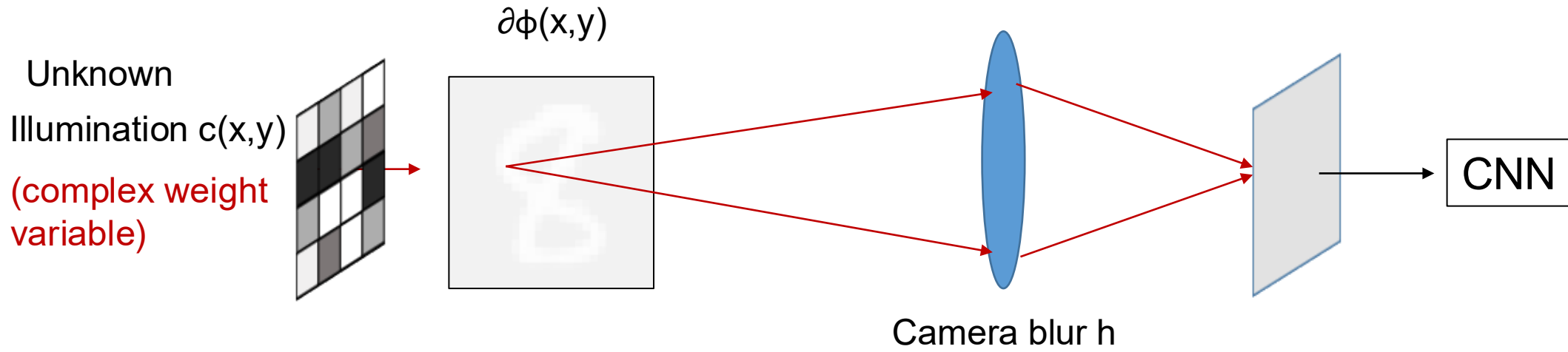


1. Normalize intensity map to 1
2. Define thickness map at some reasonable amount (100  $\mu\text{m}$  max change)
3. Convert thickness map into optical phase delay:



# Example #1: Optimizing coherent illumination pattern for improved classification

$$\text{Coherent image Model: } I_c(x,y) = |h * (c \odot \partial\phi)|^2$$



$c(x,y)$ ,  $\partial\phi(x,y)$  and  $h(x,y)$  are all complex-valued

## Example #2: Optimizing aperture shape for improved digit classification

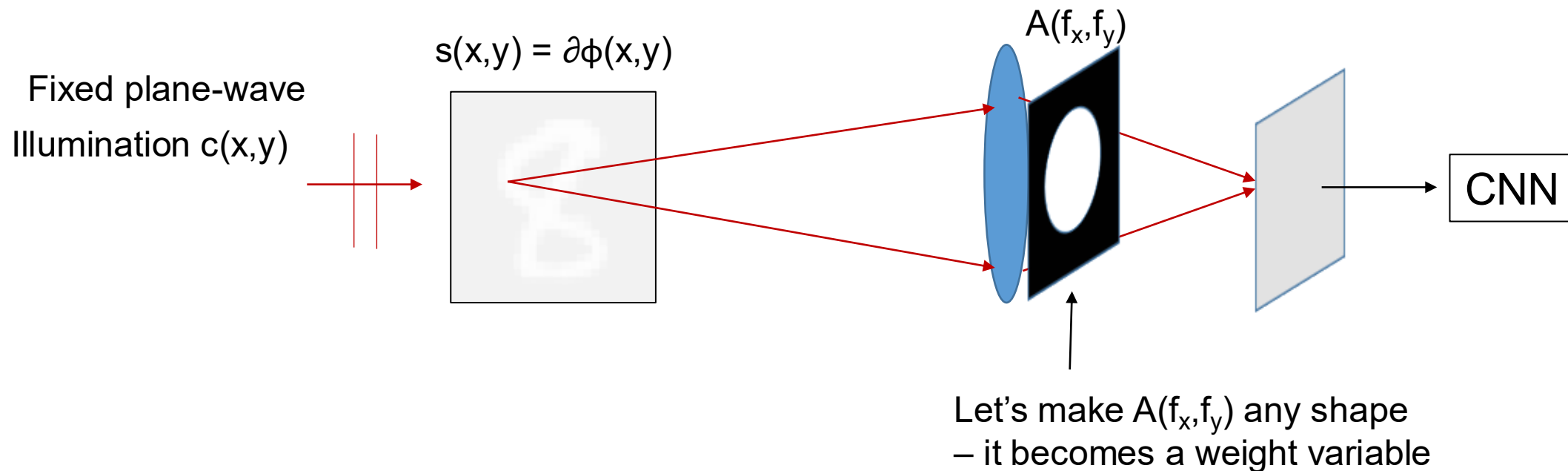
Example future situation: Hacking has brought online banking to a halt. We now rely on a special form of physical check that is made of visibly transparent plastic. To write the amount in, you press down with a pen-like instrument, and then the check is read out by shining a particular pattern of laser light onto it, and then imaging it with a lens.

Question #2: What type of aperture shape should you use to maximize classification accuracy?

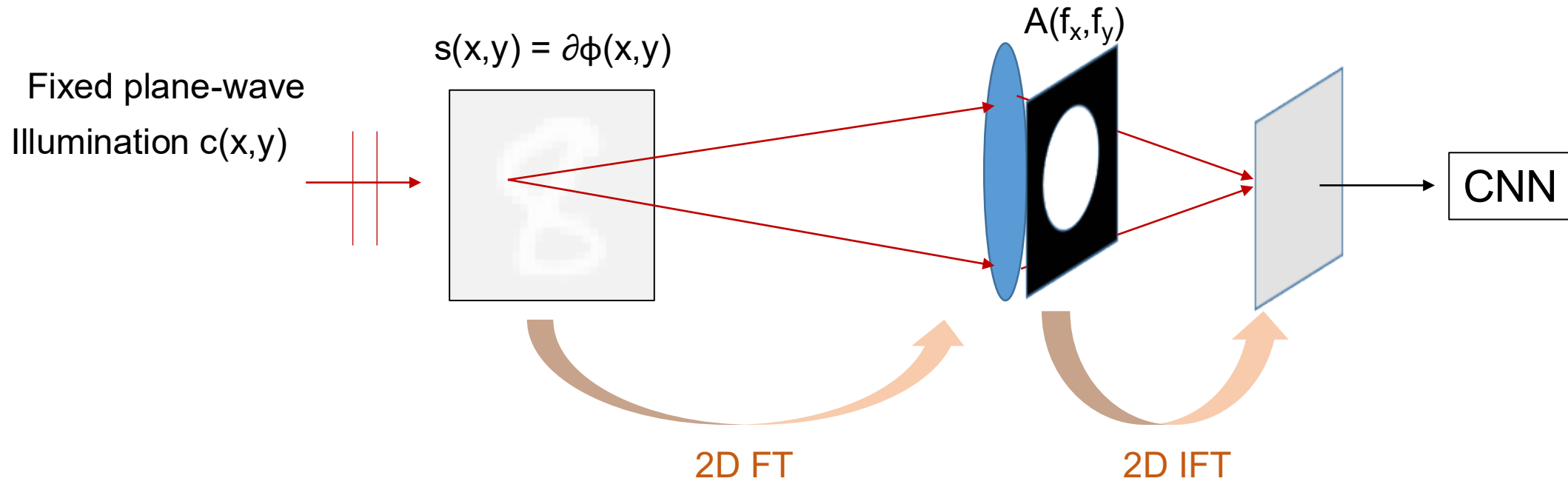
## Example #2: Optimizing aperture shape for improved digit classification

Example future situation: Hacking has brought online banking to a halt. We now rely on a special form of physical check that is made of visibly transparent plastic. To write the amount in, you press down with a pen-like instrument, and then the check is read out by shining a particular pattern of laser light onto it, and then imaging it with a lens.

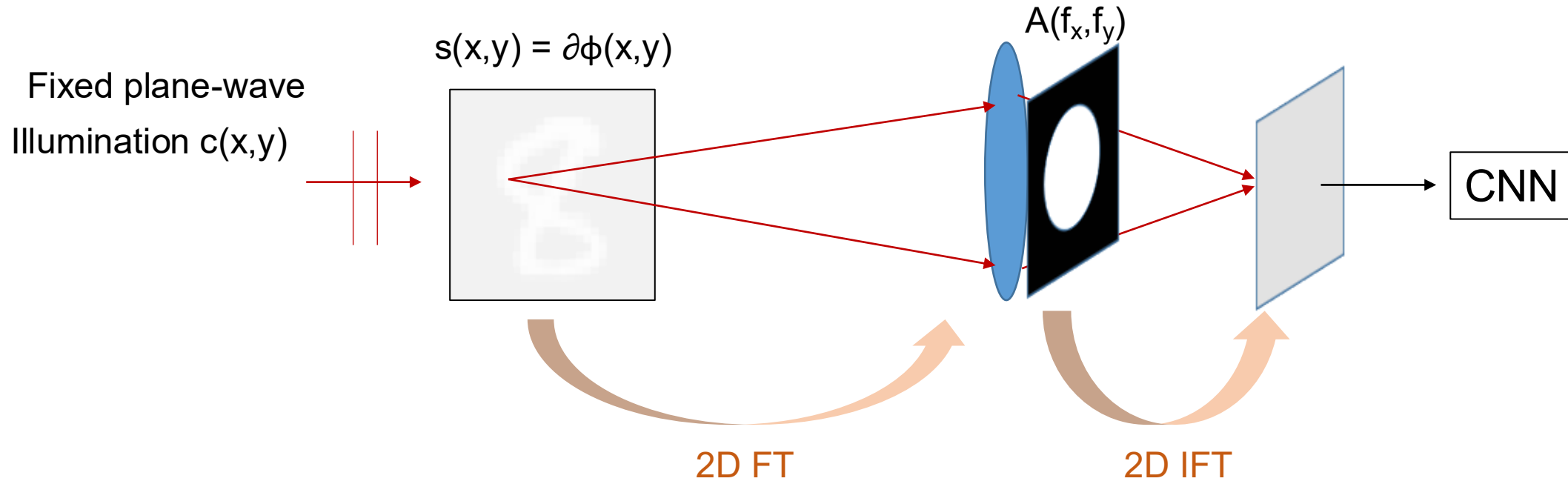
Question #2: What type of aperture shape should you use to maximize classification accuracy?



## Example #2: Optimizing aperture shape for improved digit classification

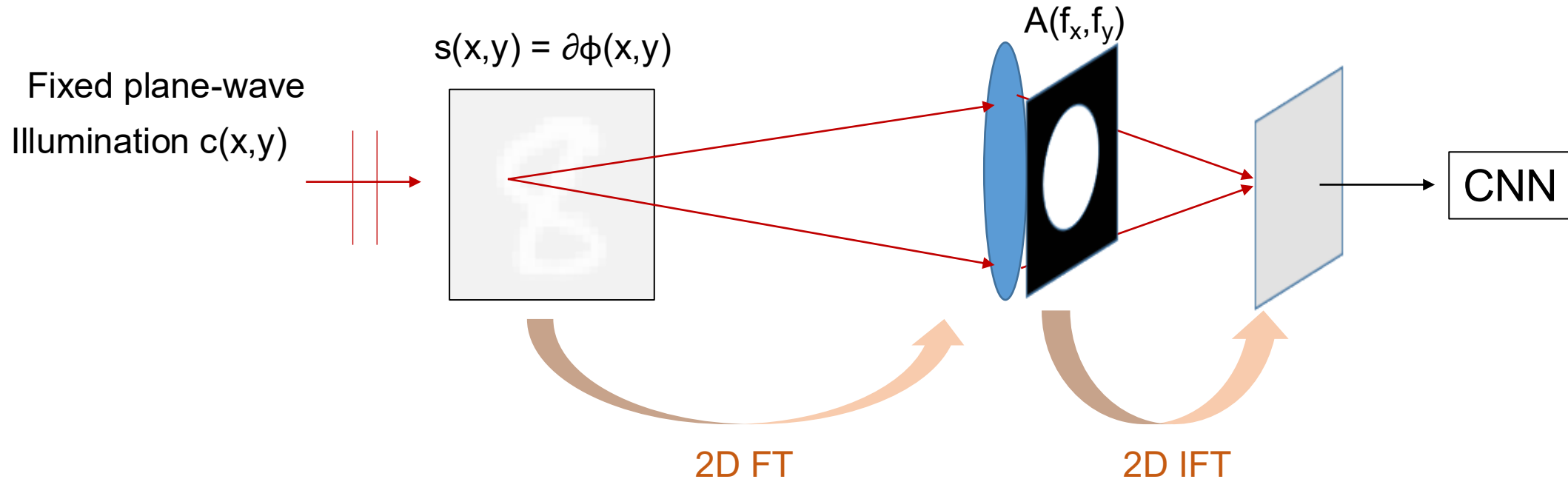


## Example #2: Optimizing aperture shape for improved digit classification



$$\text{Coherent image Model: } I_c(x,y) = |F^{-1}[A \odot F[c \odot \partial\phi]]|^2$$

## Example #2: Optimizing aperture shape for improved digit classification



$$\text{Coherent image Model: } I_c(x,y) = |F^{-1}[A \odot F[c \odot \partial\phi]]|^2$$

$$\text{For plane wave: } C(x,y) = 1$$

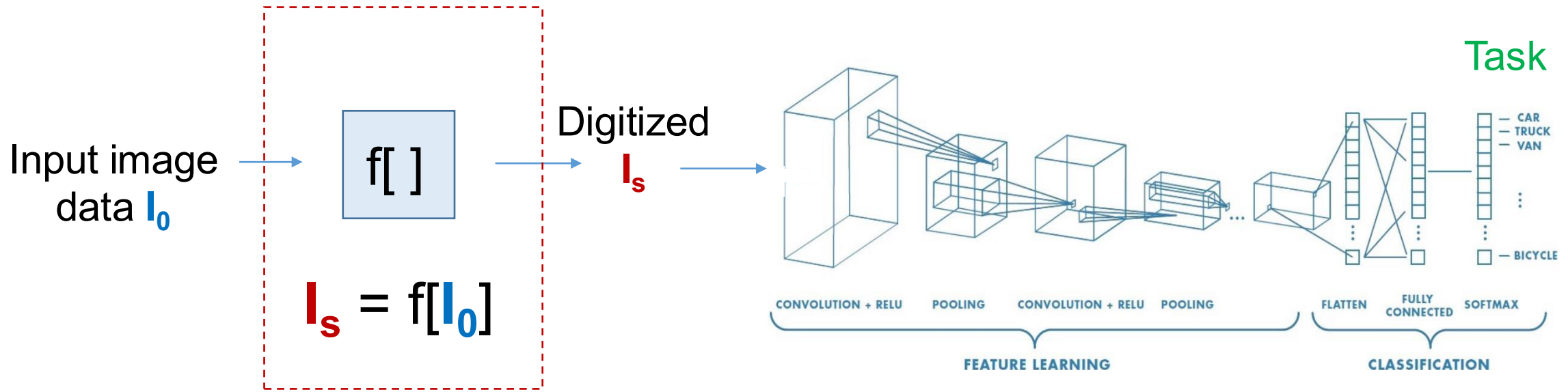
$$\text{Coherent image Model: } I_c(x,y) = |F^{-1}[A \odot F[\partial\phi]]|^2$$

## Remaining questions to address about physical layers:

- Where and how should I implement my physical layer?
  - Simulation data
  - Experimental data
- How can I add some constraints to the physical weights that I'm optimizing?
- What are some common issues and pitfalls?

## Physical Layers

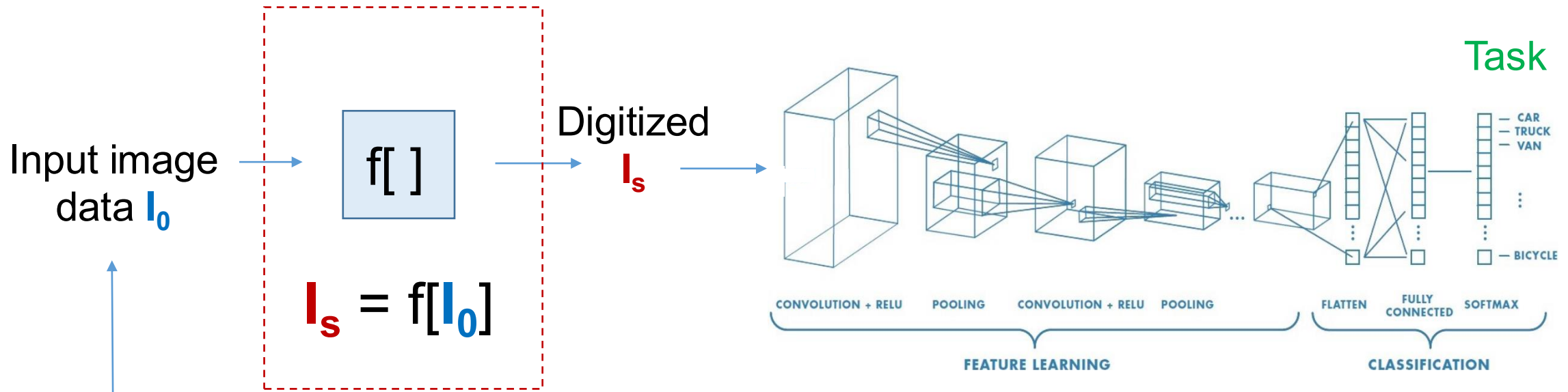
## Digital Layers



Q: Where and how should I implement my physical layer?

## Physical Layers

## Digital Layers



Q: Where and how should I implement my physical layer?

A: It depends on your data and implementation

- Situation #1: Fully simulated physical layers
- Situation #2: Experimentally-driven physical layers

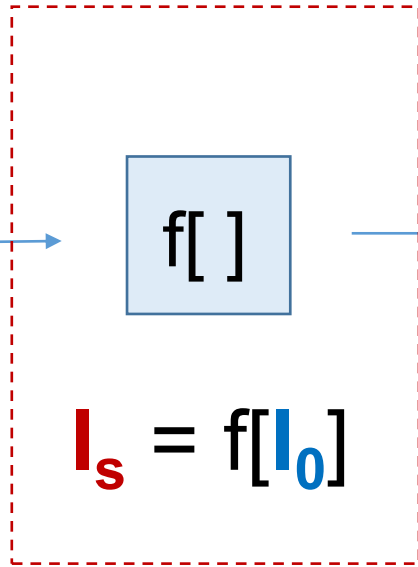
# Situation #1: Fully simulated physical layers

Option (a): Simulate the input images and the labels from scratch



Full simulation of different samples and their labels

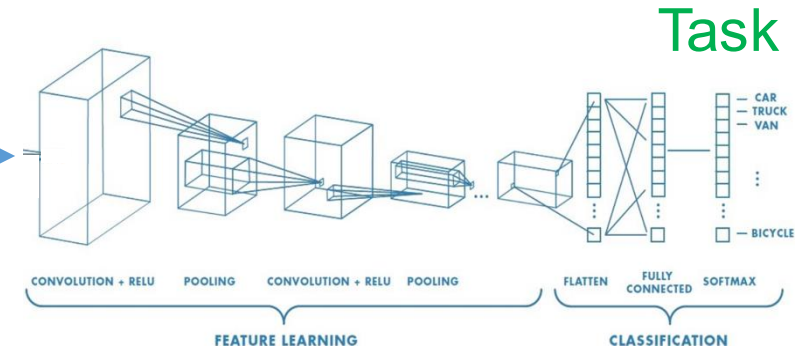
## Physical Layers



Digitized

$I_s$

## Digital Layers



Simulated labels

# Situation #1: Fully simulated physical layers

Option (a): Simulate the input images and the labels from scratch

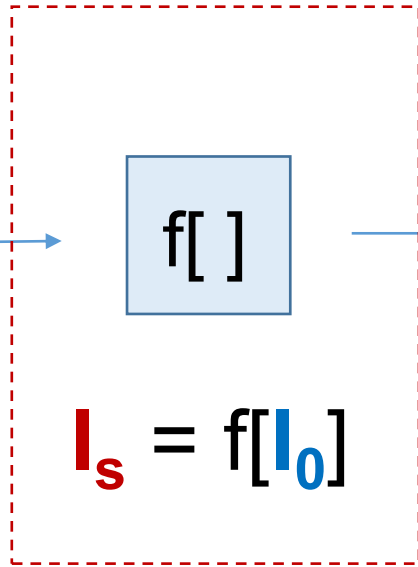


Full simulation of different samples and their labels

Examples:

- [Ultrasound scatterers, segmentation boundary]
- [Simulated cell body types, location]
- [CT phantom, 3D mesh surfaces]

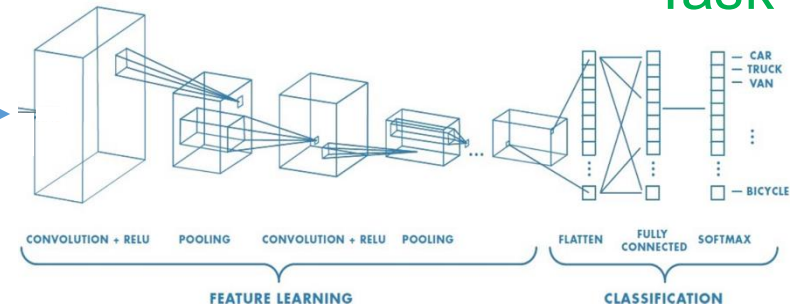
## Physical Layers



Digitized

$I_s$

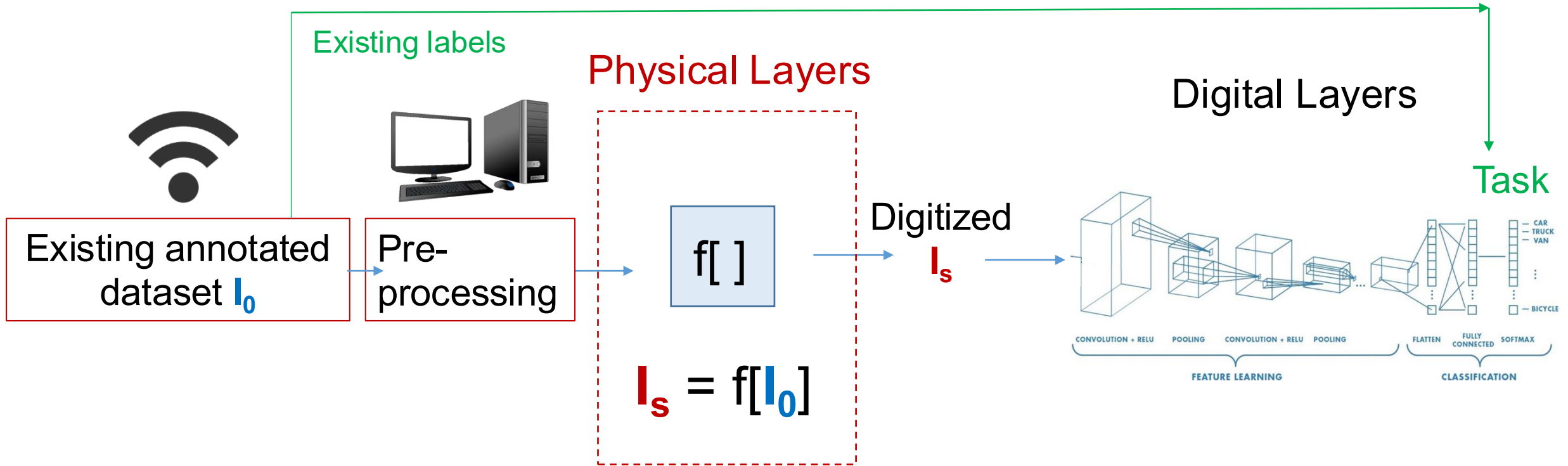
## Digital Layers



Simulated labels

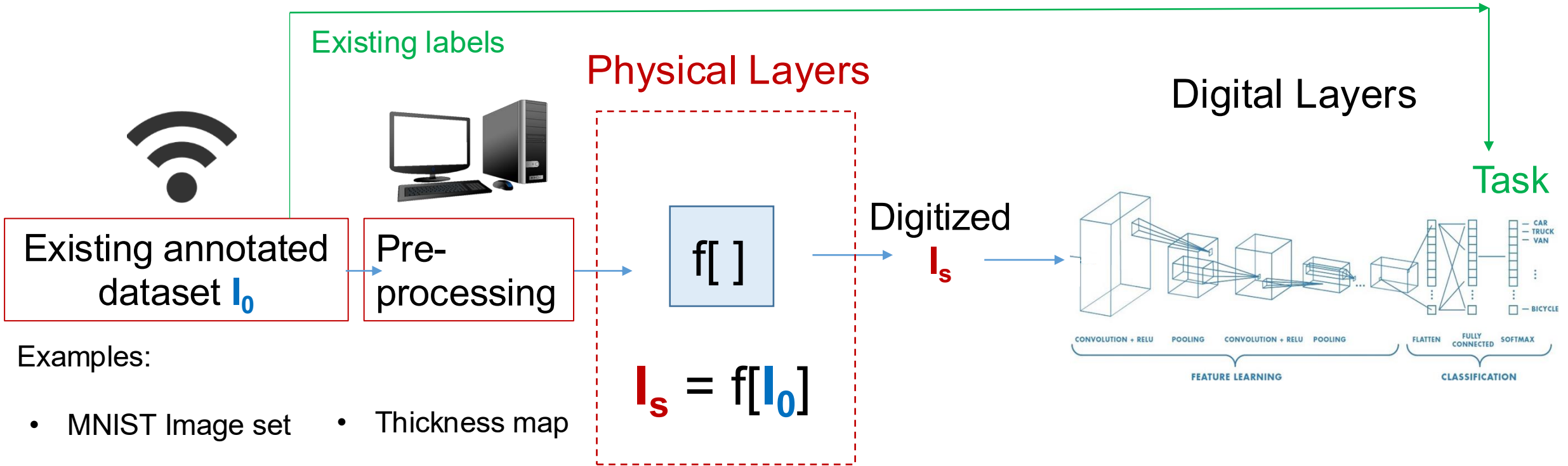
# Situation #1: Fully simulated physical layers

Option (b): Augment an existing dataset that you download



# Situation #1: Fully simulated physical layers

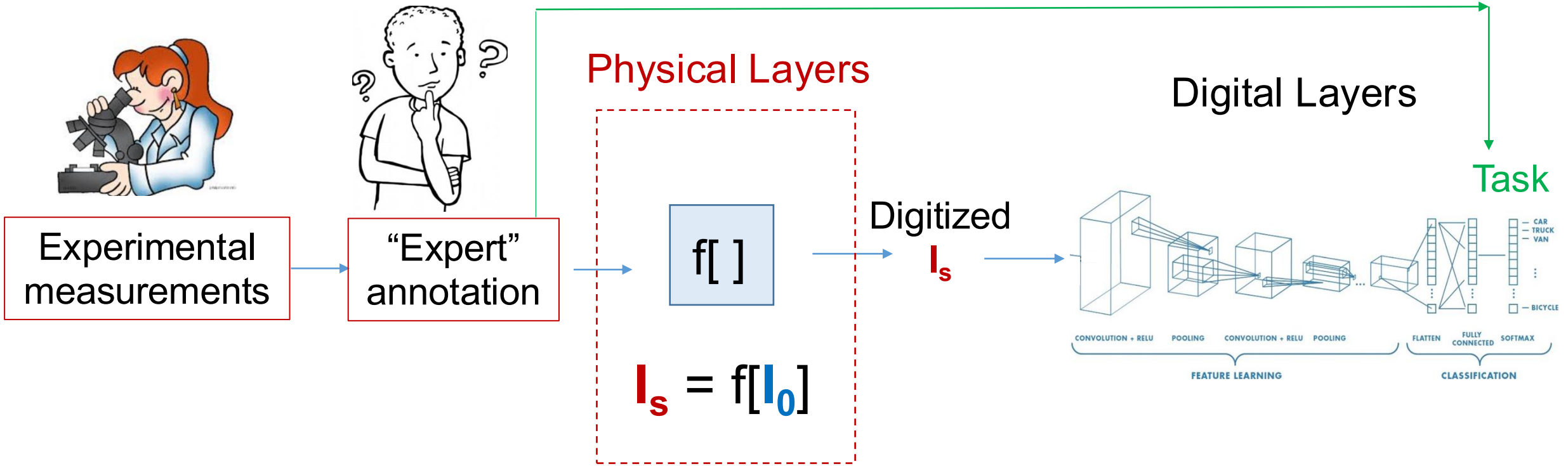
Option (b): Augment an existing dataset that you download



Examples:

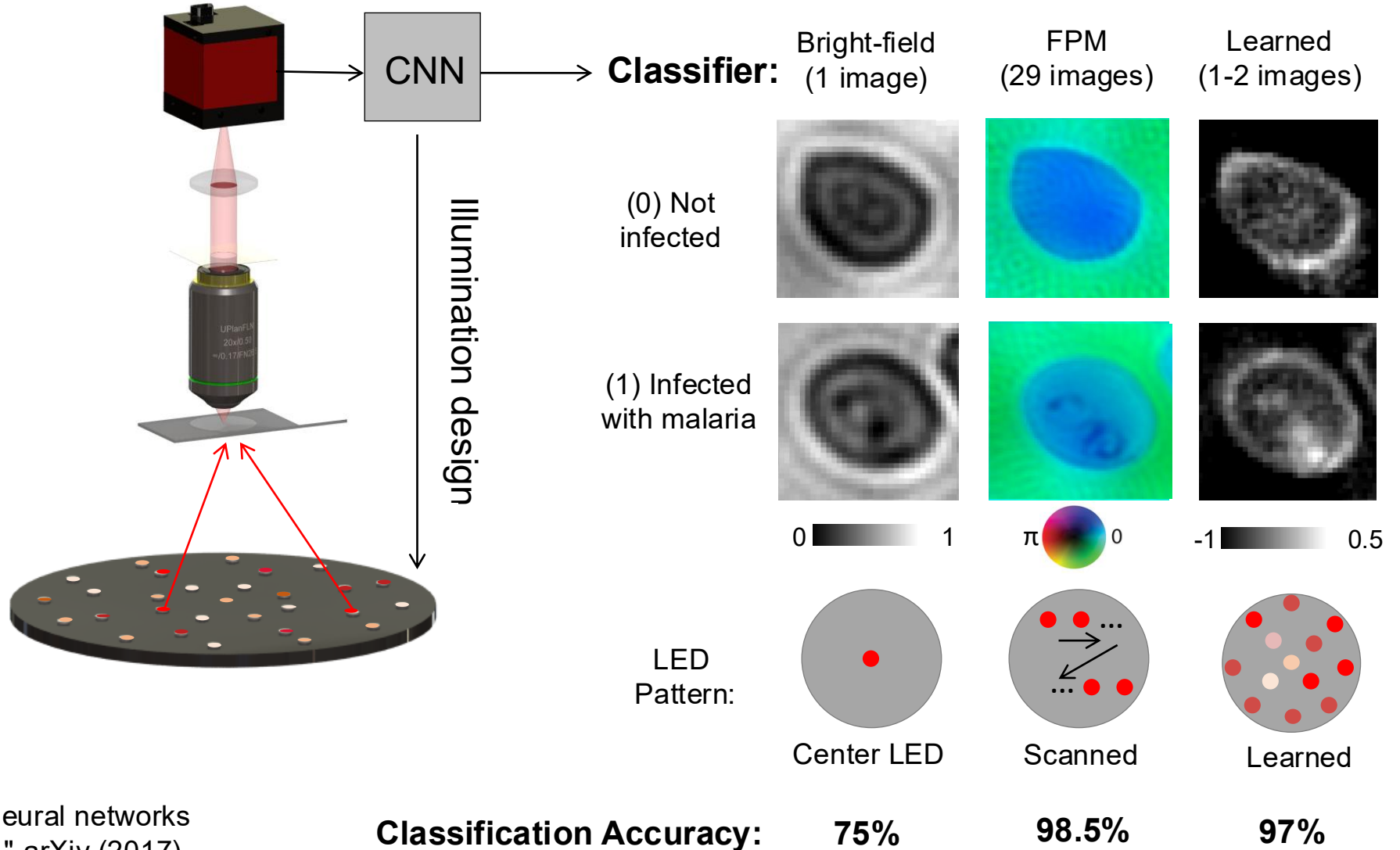
- MNIST Image set
- Segmented cells from Celltracker
- Segmented CT dataset from lab
- Thickness map
- Multispectral image stack
- Stitch together in a 3D composite

# Situation #2: Experimentally-driven physical layers



# Situation #2: Experimentally-driven physical layers

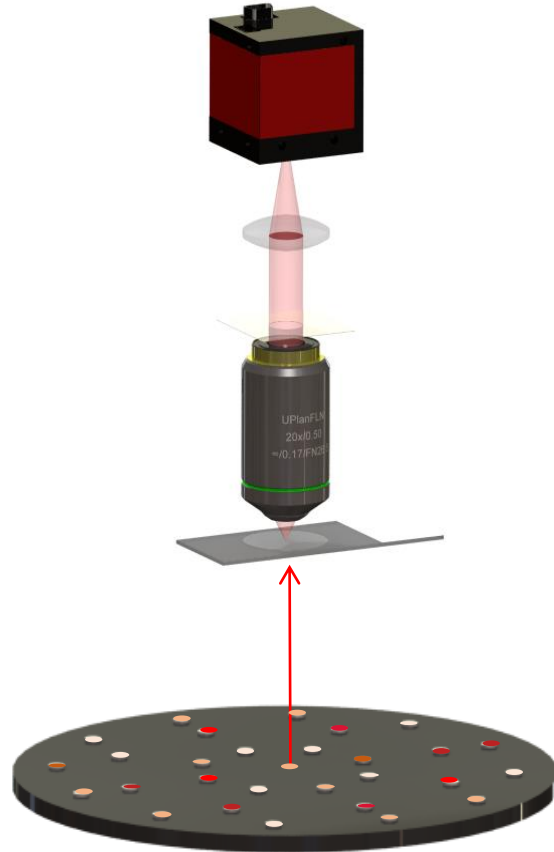
Example: CNN-Optimized illumination for classification of malaria:



R. Horstmeyer et al., "Convolutional neural networks that teach microscopes how to image," arXiv (2017)

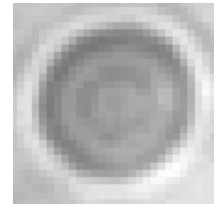
## Situation #2: Experimentally-driven physical layers

Example: CNN-Optimized illumination for classification of malaria:



Data set for physical layer optimization:

Turn on LED 1, capture image 1:



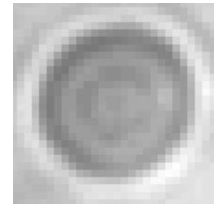
## Situation #2: Experimentally-driven physical layers

Example: CNN-Optimized illumination for classification of malaria:

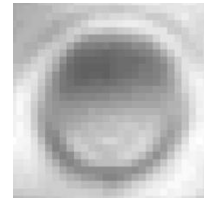


Data set for physical layer optimization:

Turn on LED 1, capture image 1:



Turn on LED 1, capture image 2:



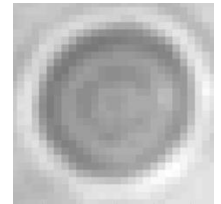
## Situation #2: Experimentally-driven physical layers

Example: CNN-Optimized illumination for classification of malaria:

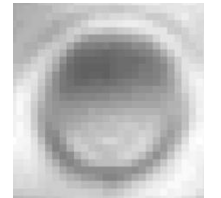


Data set for physical layer optimization:

Turn on LED 1, capture image 1:

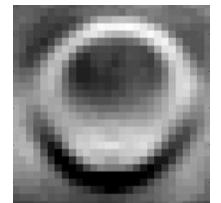


Turn on LED 1, capture image 2:



⋮

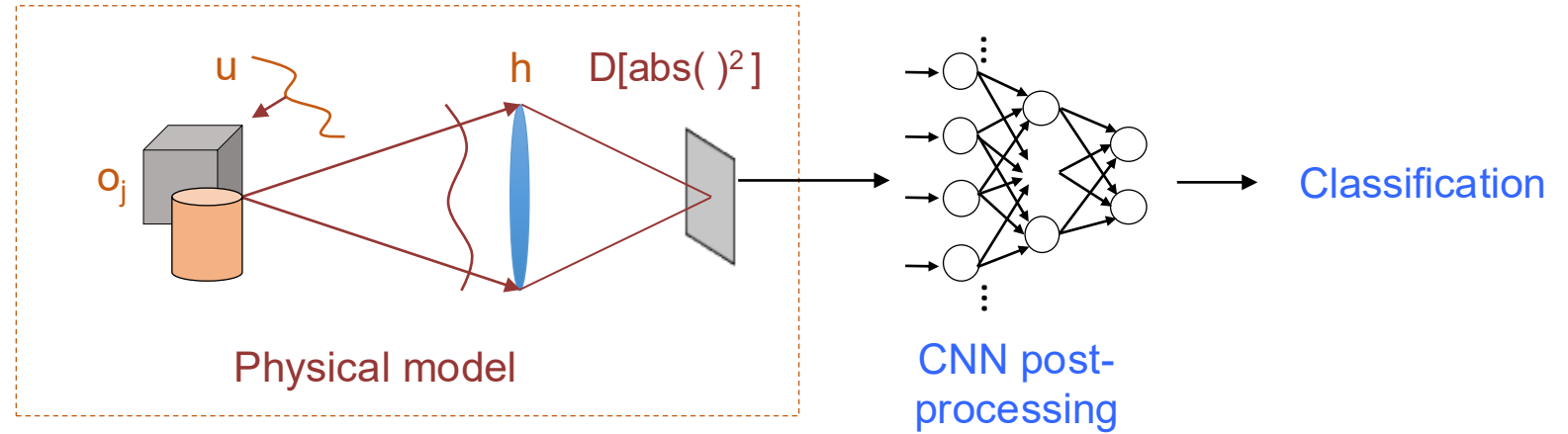
Turn on LED 32, capture image 32:



Save all 32 images (96 with 3 colors)

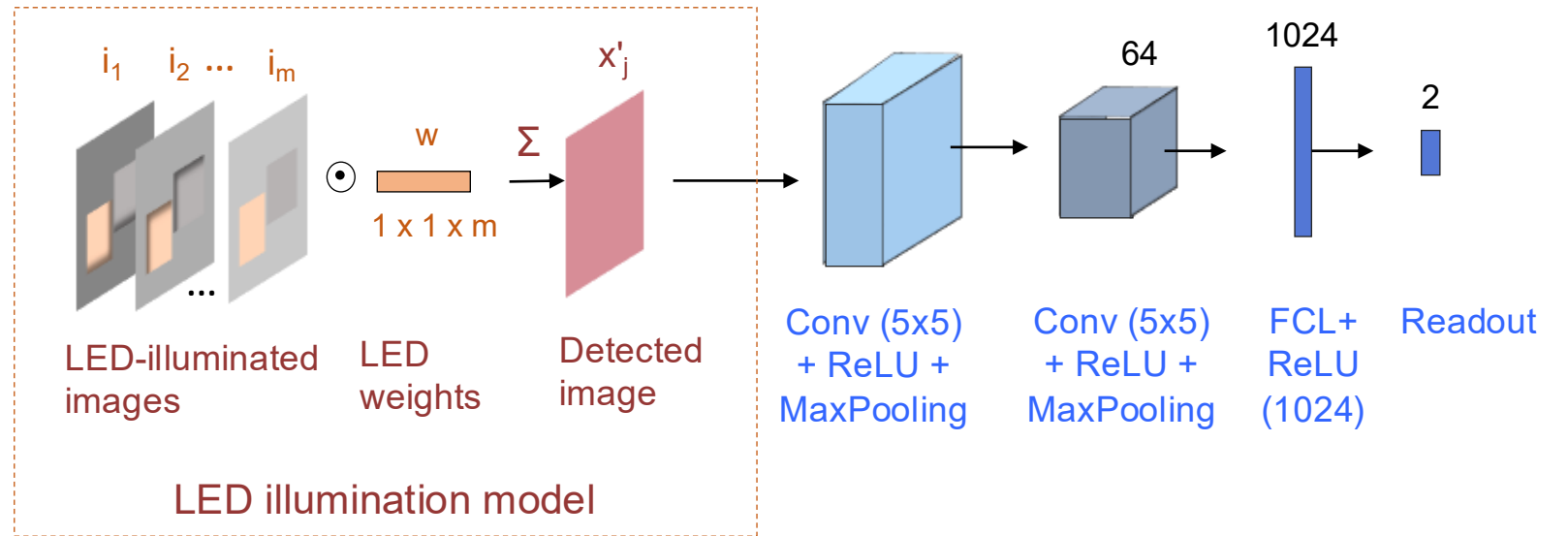
# Situation #2: Experimentally-driven physical layers

Example: CNN-Optimized illumination for classification of malaria:

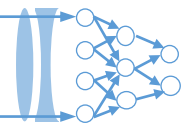


Physical layer:

$$I_s = \sum w_j I_j$$

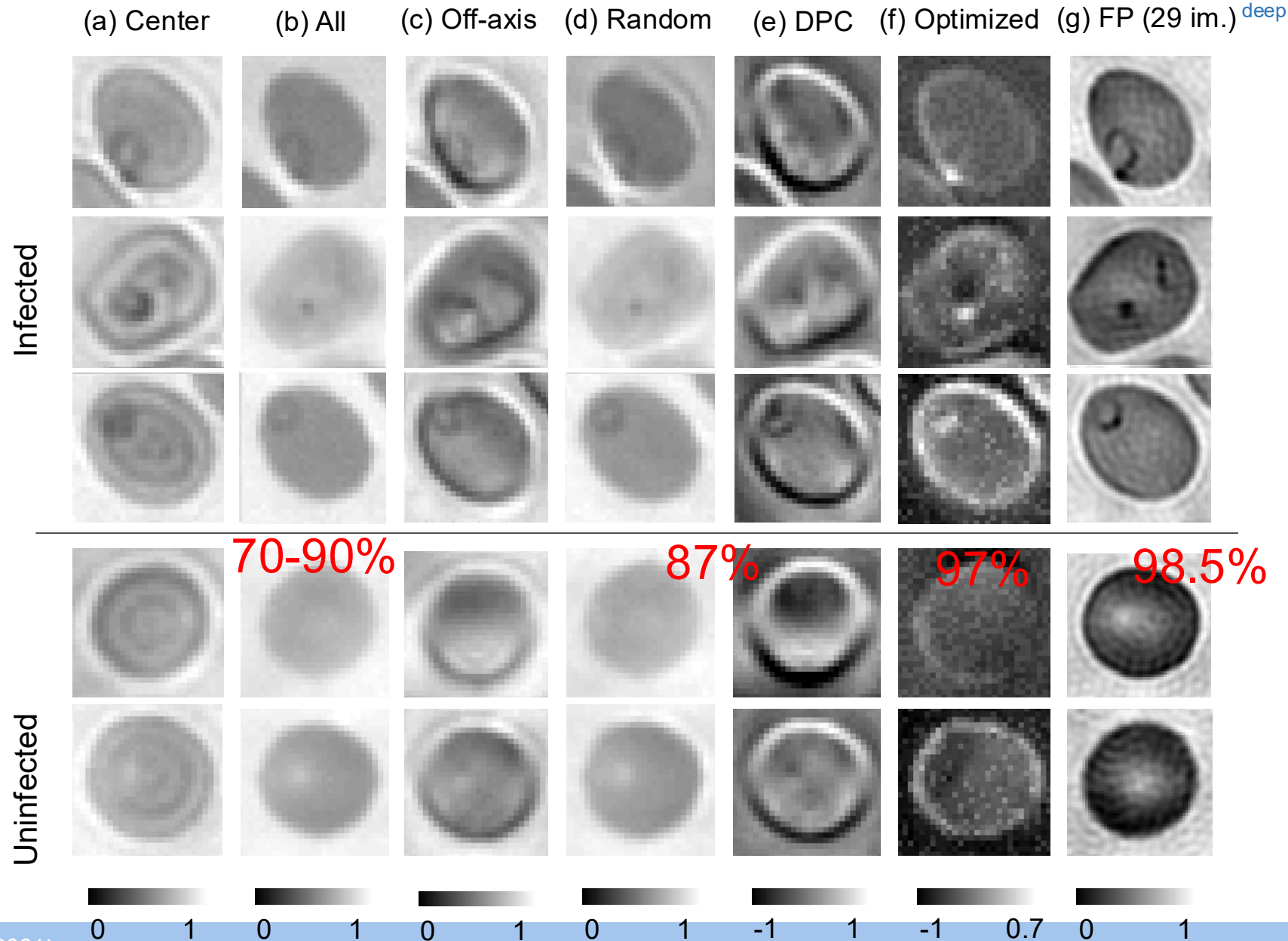


# Situation #2: Experimentally-driven physical layers



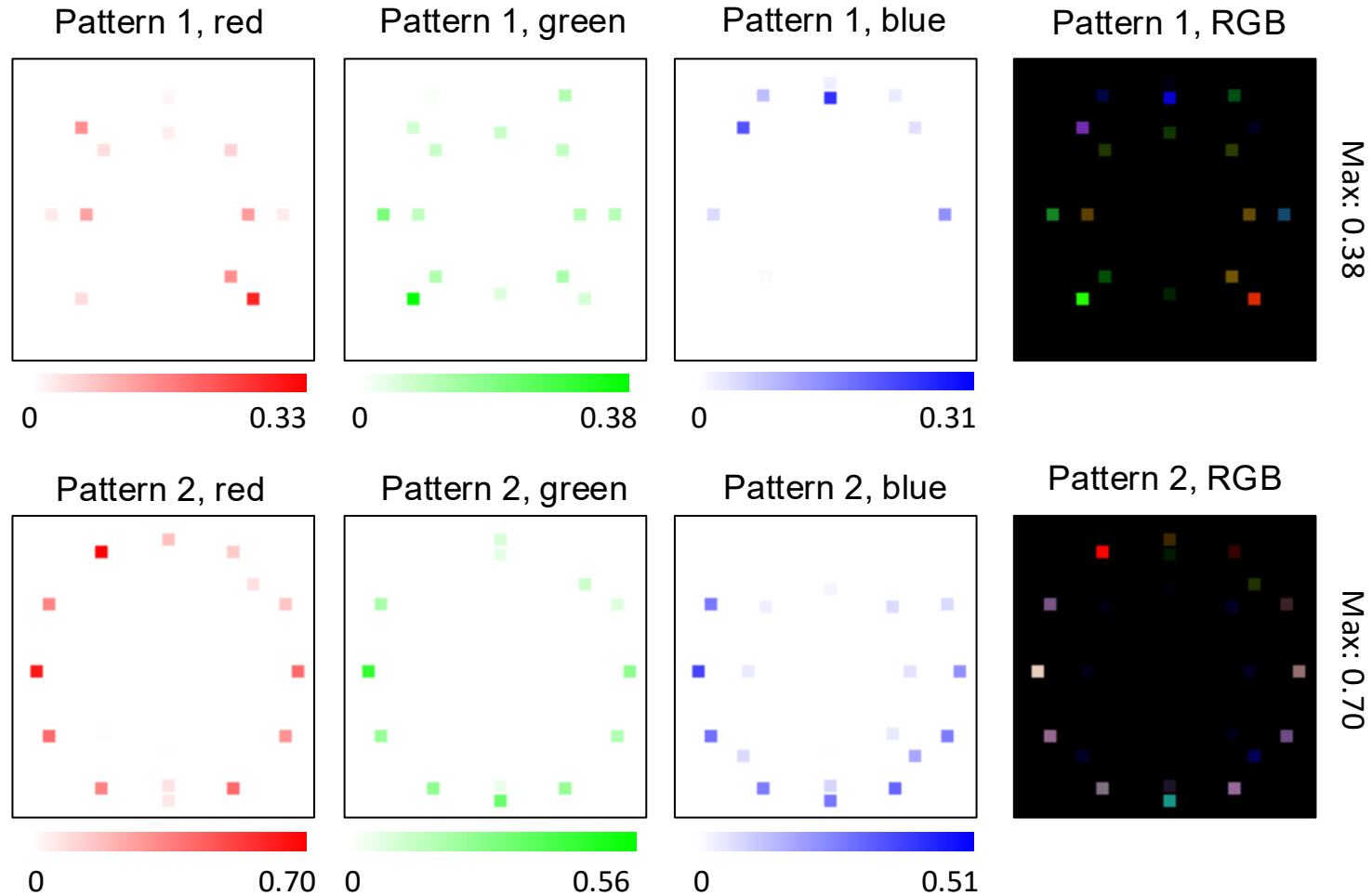
deep imaging

Example: CNN-Optimized illumination for classification of malaria:

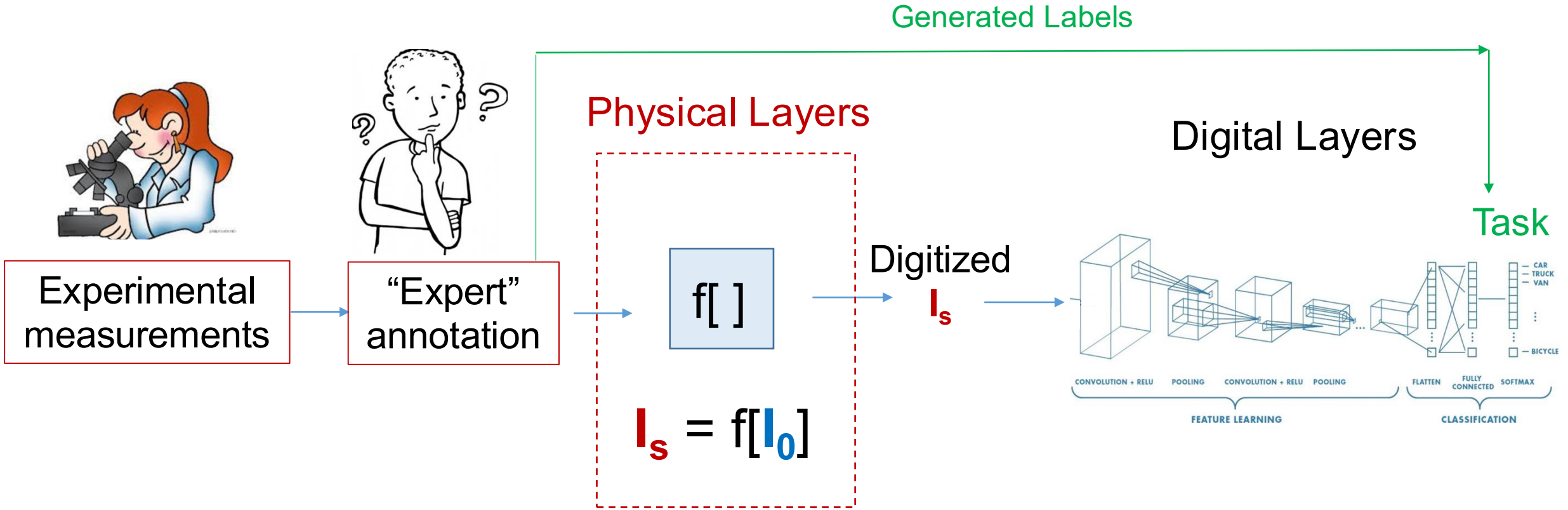


# Situation #2: Experimentally-driven physical layers

## Optimized color LED patterns to classify malaria



# Situation #2: Experimentally-driven physical layers



Pro's of experimental measurements: Don't need to worry about making your simulations match the setup! (HUGE WIN)

Con's of experimental measurements: You'll need to label them, limited access to desired sample information, often need to exploit some fundamental physical property

# How can I add some constraints to my physical weights?

Without any constraints, weights can be any real (or complex) number. What if you physically can't realize any real or physical number?

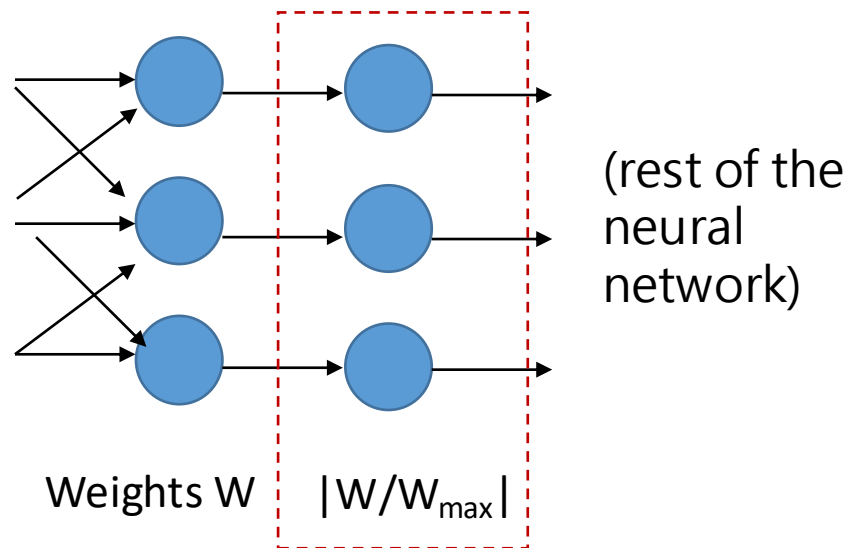
Example: Constrain weights to be non-negative values less than one

## How can I add some constraints to my physical weights?

Without any constraints, weights can be any real (or complex) number. What if you physically can't realize any real or physical number?

Example: Constrain weights to be non-negative values less than one

Solution: add constraint as an extra “differentiable” layer (operation)

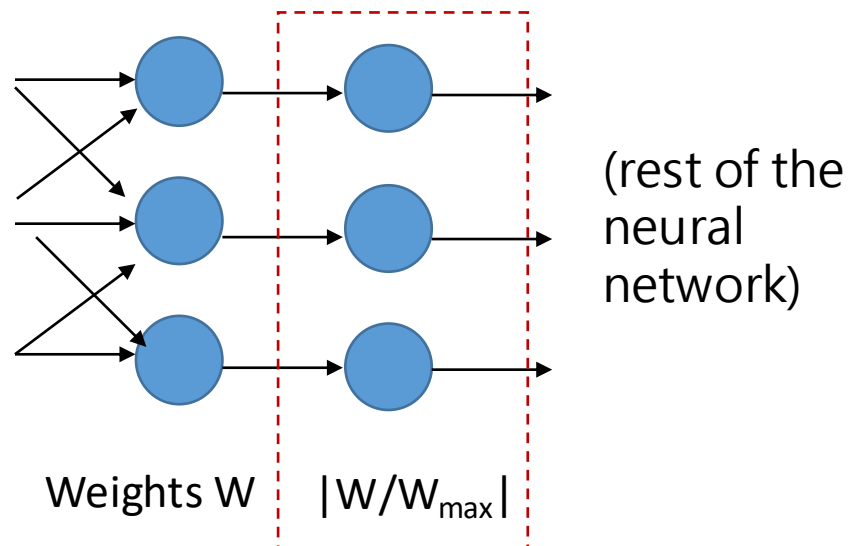


# How can I add some constraints to my physical weights?

Without any constraints, weights can be any real (or complex) number. What if you physically can't realize any real or physical number?

Example: Constrain weights to be non-negative values less than one

Solution: add constraint as an extra "differentiable" layer (operation)



Pros:

- Easy to implement
- Constraints are obvious

Cons:

- Not always a well-behaved derivative

# How can I add some constraints to my physical weights?

Without any constraints, weights can be any real (or complex) number. What if you physically can't realize any real or physical number?

Example: Constrain weights to be either 0 or 1

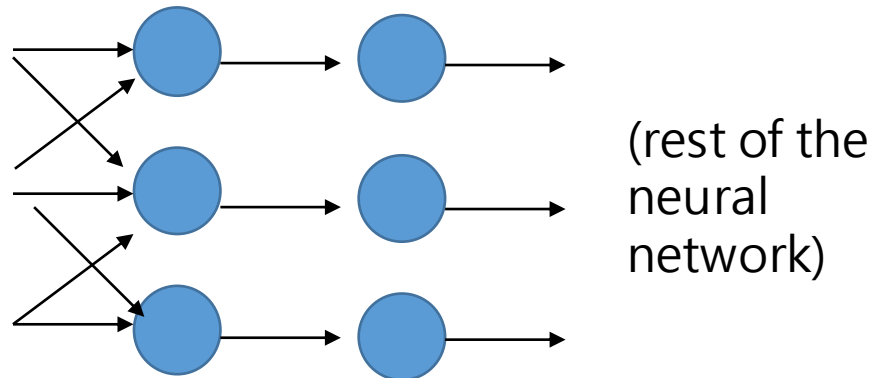
Solution: Perform annealing with a temperature parameter

# How can I add some constraints to my physical weights?

Without any constraints, weights can be any real (or complex) number What if you physically can't realize any real or physical number?

Example: Constrain weights to be either 0 or 1

Solution: Perform annealing with a temperature parameter



Weights  $W$      $I(n) = \text{Soft-max} [\alpha_t w(n)]$

Increase  $\alpha$  with iteration number

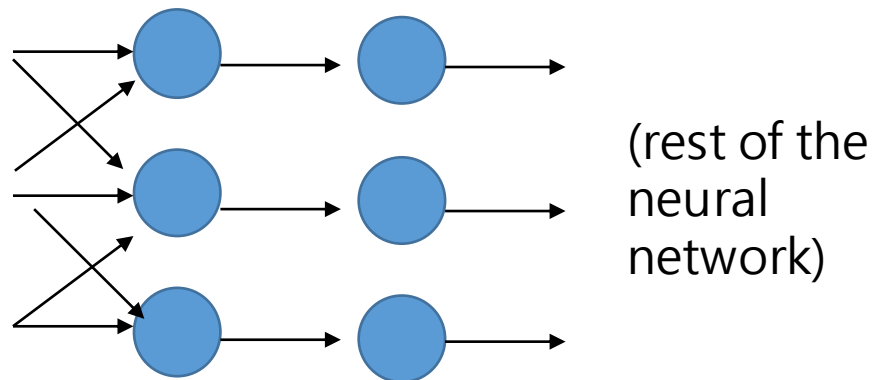
$\text{Soft-max}(x) = \exp(-x) / \sum \exp(-x)$

# How can I add some constraints to my physical weights?

Without any constraints, weights can be any real (or complex) number. What if you physically can't realize any real or physical number?

Example: Constrain weights to be either 0 or 1

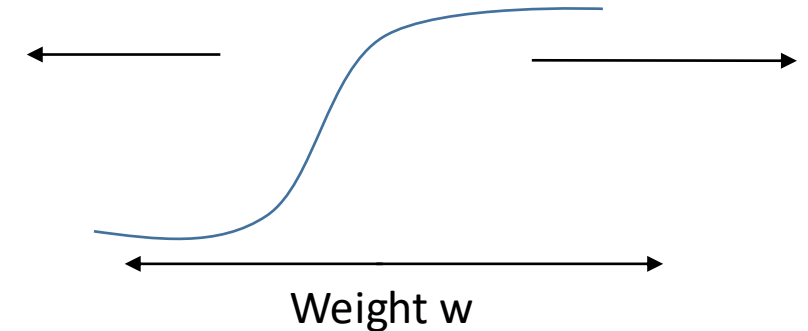
Solution: Perform annealing with a temperature parameter



Weights  $W$      $I(n) = \text{Soft-max} [\alpha_t w(n)]$

Increase  $\alpha$  with iteration number

Drive  $w$  to be large, so  $\text{softmax}(w) \rightarrow 0$  or  $1$



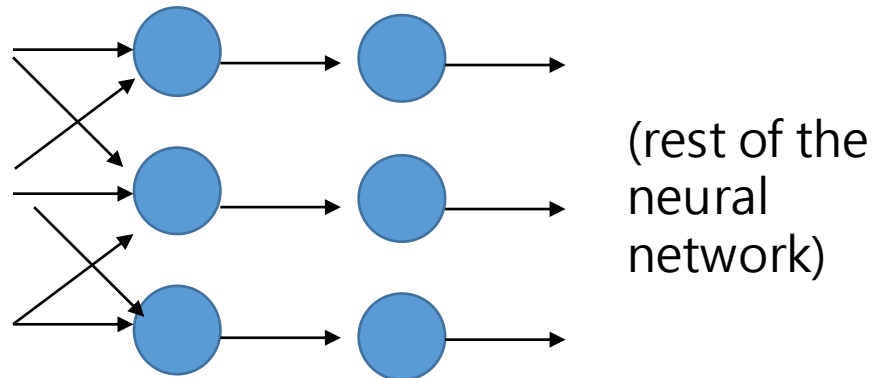
$\text{Soft-max}(x) = \exp(-x) / \sum \exp(-x)$

# How can I add some constraints to my physical weights?

Without any constraints, weights can be any real (or complex) number What if you physically can't realize any real or physical number?

Example: Constrain weights to be either 0 or 1

Solution: Perform annealing with a temperature parameter



Pros:

- Works pretty well
- Flexibly address convergence issues

Cons:

- A bit sensitive

Weights  $W$      $I(n) = \text{Soft-max} [\alpha_t w(n)]$

Increase  $\alpha$  with iteration number

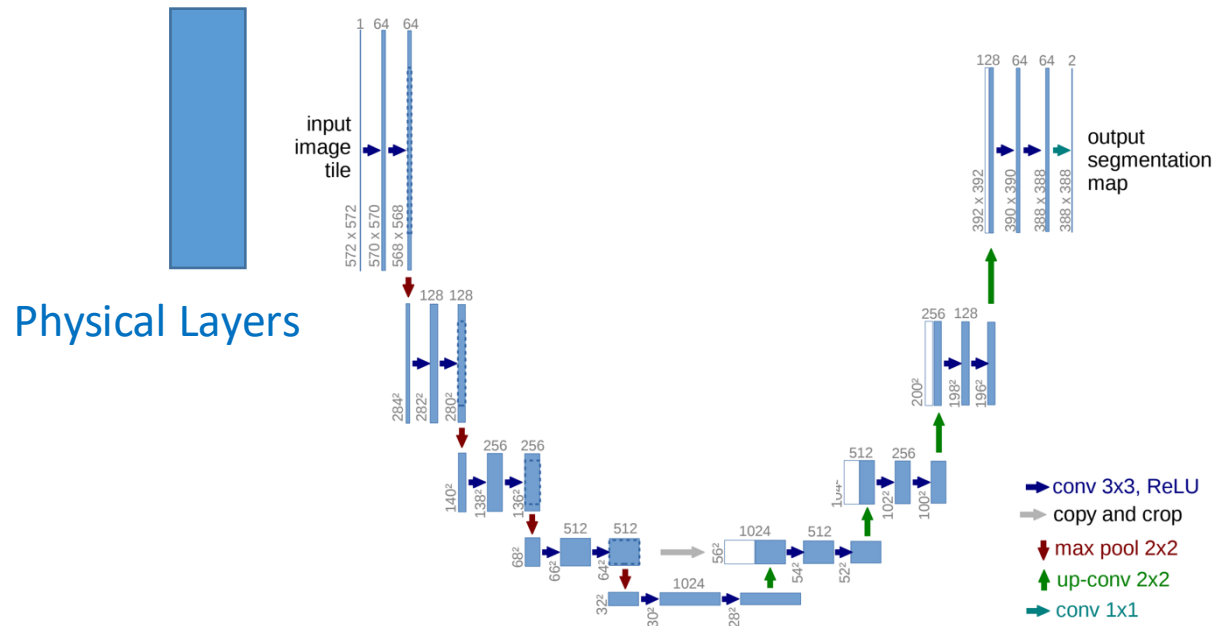
$$\text{Soft-max}(x) = \exp(-x) / \sum \exp(-x)$$

## What are some common issues and pitfalls with physical layers?

- Most common issue – you have a bug in your CNN!
  - Solution: “Disable “ physical layer (set to constant), and get network to work!
  - Good practice: always compare performance with and without physical layer
- Another common challenge - vanishing gradients

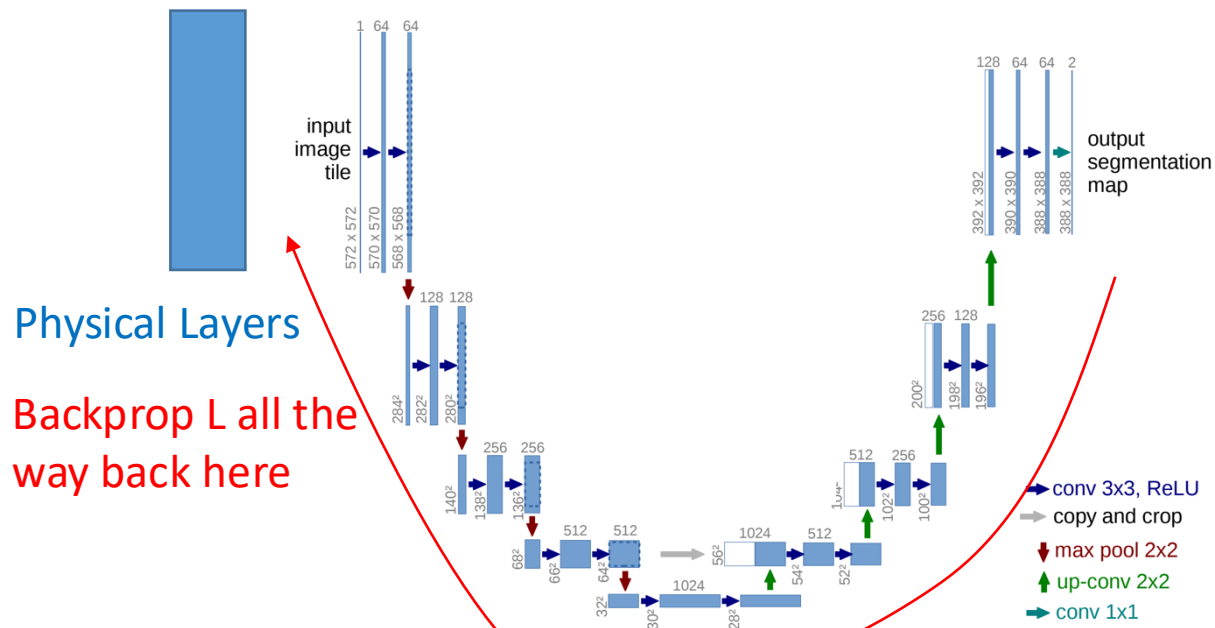
# What are some common issues and pitfalls with physical layers?

- Most common issue – you have a bug in your CNN!
  - Solution: “Disable “ physical layer (set to constant), and get network to work!
- Another common challenge - vanishing gradients



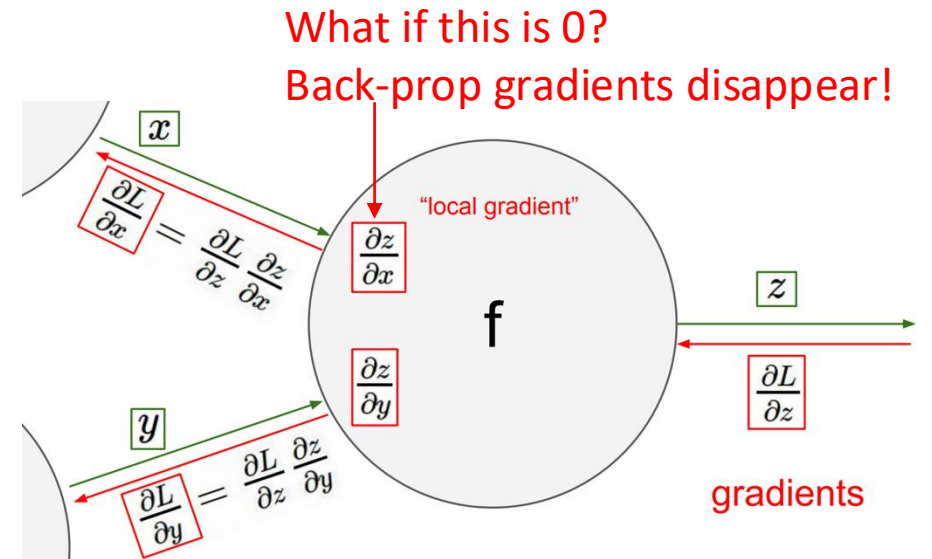
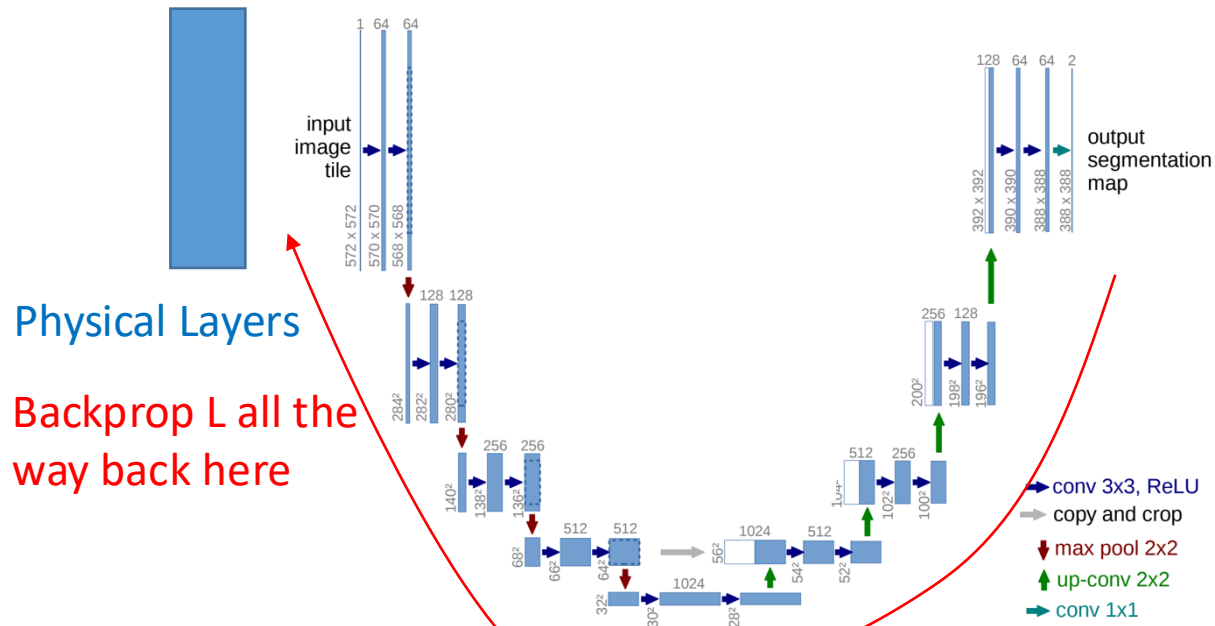
# What are some common issues and pitfalls with physical layers?

- Most common issue – you have a bug in your CNN!
  - Solution: “Disable “ physical layer (set to constant), and get network to work!
- Another common challenge - vanishing gradients



# What are some common issues and pitfalls with physical layers?

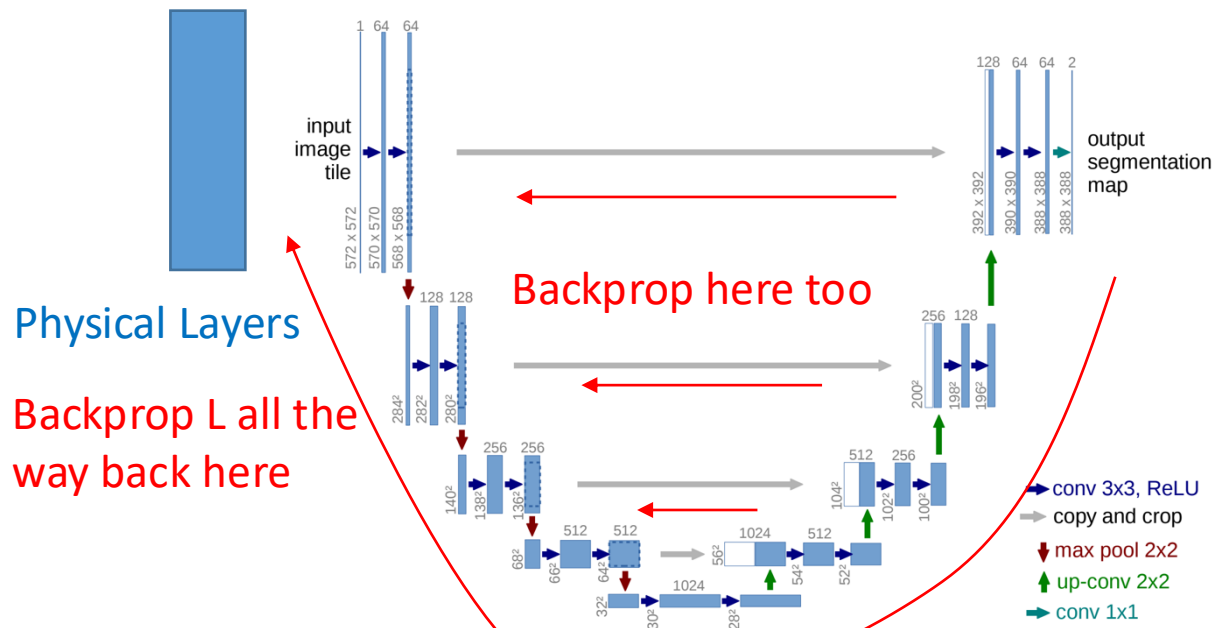
- Most common issue – you have a bug in your CNN!
  - Solution: “Disable “ physical layer (set to constant), and get network to work!
- Another common challenge - vanishing gradients



From Stanford CS231n

# What are some common issues and pitfalls with physical layers?

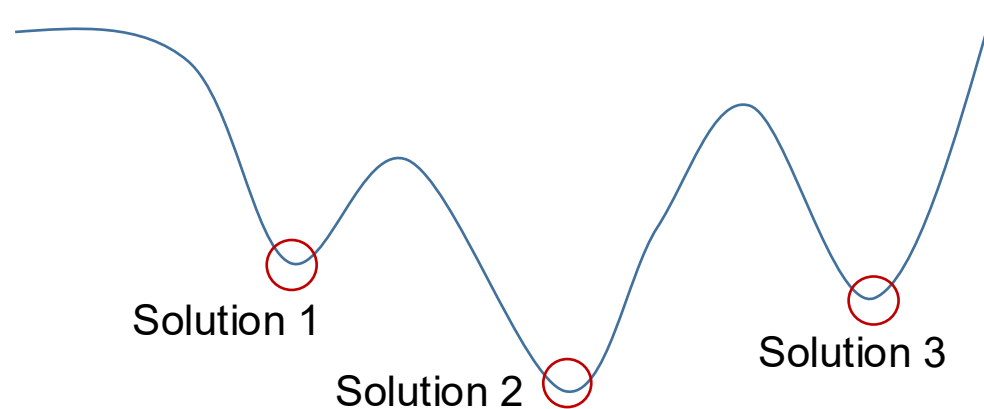
- Most common issue – you have a bug in your CNN!
  - Solution: “Disable “ physical layer (set to constant), and get network to work!
- Another common challenge - vanishing gradients



Solution: Introduce skipped connections

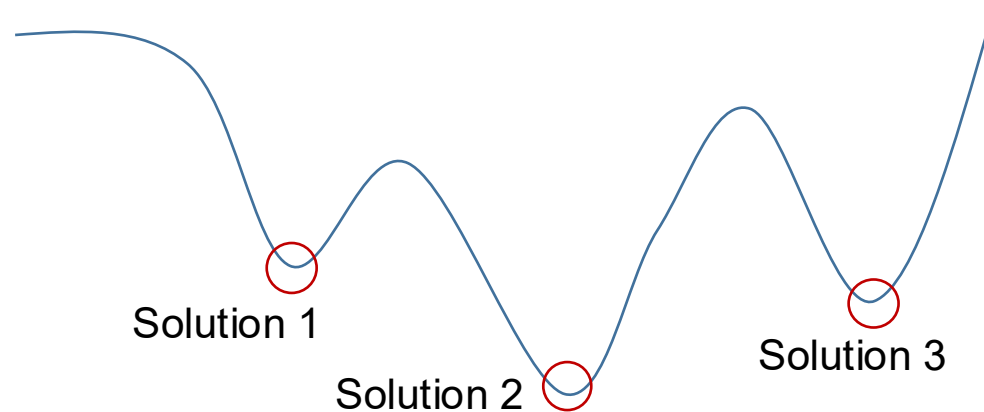
## What are some common issues and pitfalls with physical layers?

- Most common issue – you have a bug in your CNN!
  - Solution: “Disable “ physical layer (set to constant), and get network to work!
- Another common challenge - vanishing gradients
- Third issue - physical layer results are not very repeatable...



## What are some common issues and pitfalls with physical layers?

- Most common issue – you have a bug in your CNN!
  - Solution: “Disable “ physical layer (set to constant), and get network to work!
- Another common challenge - vanishing gradients
- Third issue - physical layer results are not very repeatable...



Effective Solution: Add a small amount of noise to the physical layer output:

$$I_s = \sum w_j I_j + n$$

# Ethical questions surrounding deep convolutional networks

1. What are your expectations for an image reconstruction algorithm used in a clinical setting?

# Ethical questions surrounding deep convolutional networks

1. What are your expectations for an image reconstruction algorithm used in a clinical setting?
2. What types of “guarantees” should we be able to make, if any, to a patient?

# Ethical questions surrounding deep convolutional networks

1. What are your expectations for an image reconstruction algorithm used in a clinical setting?
2. What types of “guarantees” should we be able to make, if any, to a patient?
3. How should we guide future development of ML software to meet any guarantees?

# Ethical questions surrounding deep convolutional networks

1. What are your expectations for an image reconstruction algorithm used in a clinical setting?
2. What types of “guarantees” should we be able to make, if any, to a patient?
3. How should we guide future development of ML software to meet any guarantees?
4. How should we guide future development of ML-designed hardware to meet any guarantees?
5. Thoughts towards a system of checks and balances?

## The Machine Learning in Imaging Ethics Questionnaire

Situation 1: In 5 years, you walk into a clinic because you have a spot on your skin that you are concerned about. The clinician is too busy, so you step over to a terminal with a standard microscope and it images your arm. It says you are fine.

Are you comfortable with leaving the office?

Yes:

No:

Why or why not? What might change how you feel?

## The Machine Learning in Imaging Ethics Questionnaire

Situation 2: The same thing happens. But this time, the machine reports that it is 99% confident in its diagnosis, given previous examples of skin marks that have been verified by doctors as benign. It also gives you the opportunity to take a look at some of these previous example images it is basing its decision on. You notice that they don't look 100% like the mark on your arm, as is expected, but they look pretty similar.

Are you comfortable with leaving the office?

Yes:

No:

Why or why not? What might change how you feel?

## The Machine Learning in Imaging Ethics Questionnaire

Situation 3: In 5 years, the same thing happens. But this time, a doctor comes up after the machine makes its suggested diagnosis. He takes a very cursory look (10 seconds) and then confirms the machine's opinion.

Are you now comfortable with leaving the office?

Yes:

No:

## The Machine Learning in Imaging Ethics Questionnaire

Situation 4: In 10 years, you go up to a modified microscope, “the Tissue Scanner 3000”, that has a number of fancy lenses and lights. As a machine learning expert by now, you’re aware that this microscope is optimized for looking at skin lesions. It performs a scan with a particular lighting configuration and reports a score of 98% confident that the lesion is benign, allowing you to look through other examples. It asks If you’d like another scan for additional confidence or a different outcome, at which point the illumination changes and it does some more scanning and reports a 99% confidence level. You can continue with another scan, but...

Are you now comfortable with leaving the office?

Yes:

No: