Learning for vision

Big questions:

1. How do you represent the input and output?

2. What is the objective?

3. What is the hypothesis space? (e.g., linear, polynomial, neural net?)

4. How do you optimize? (e.g., gradient descent, Newton's method?)

5. What data do you train on?

Image colorization

1. How do you represent the input and output?

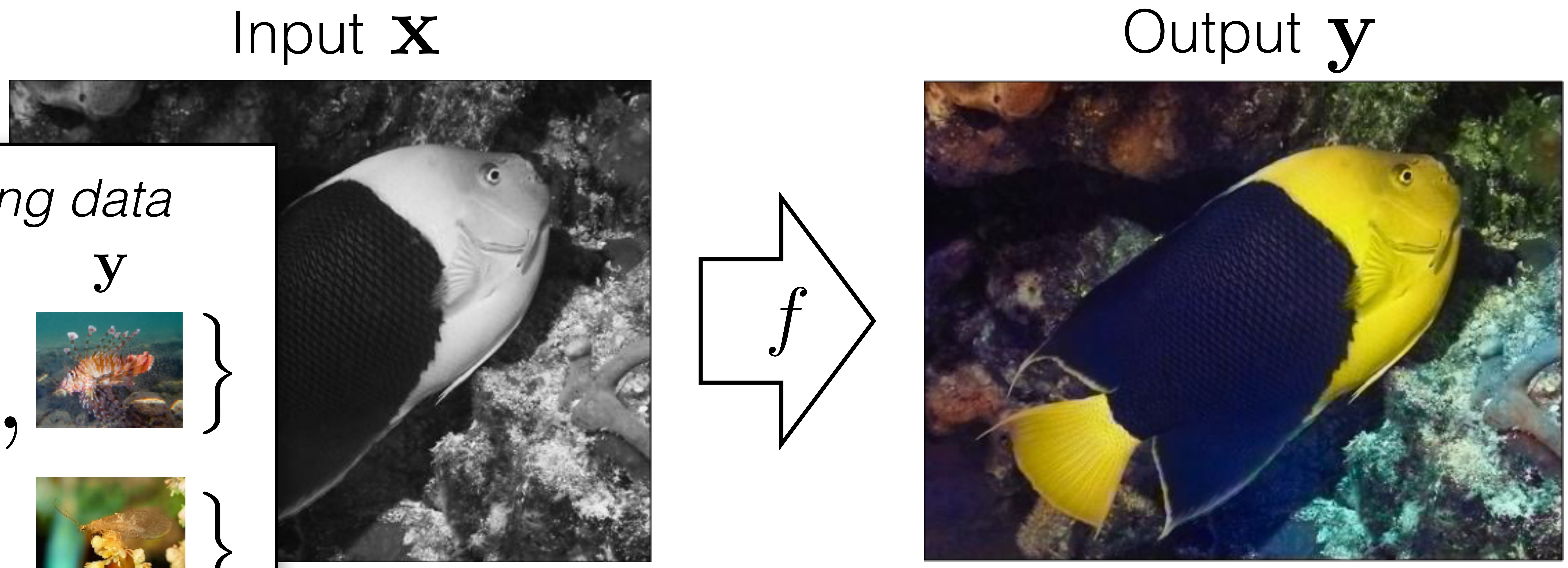
2. What is the objective?

3. Assume hypothesis space is sufficienly expressive

4. Assume we optimize perfectly

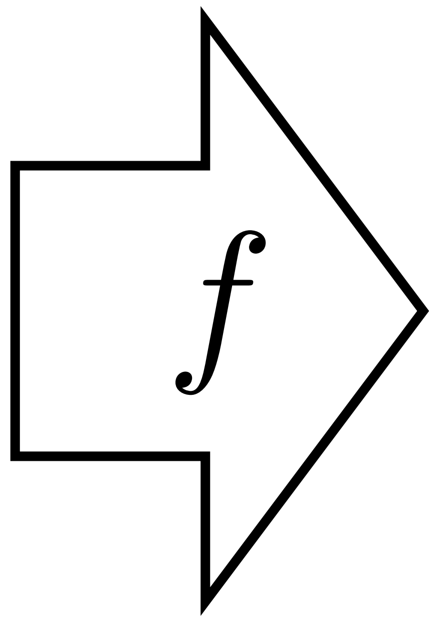
5. **What data do we train on?**

Image colorization



$$\arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}_i), \mathbf{y}_i)$$

[Zhang et al., ECCV 2016]



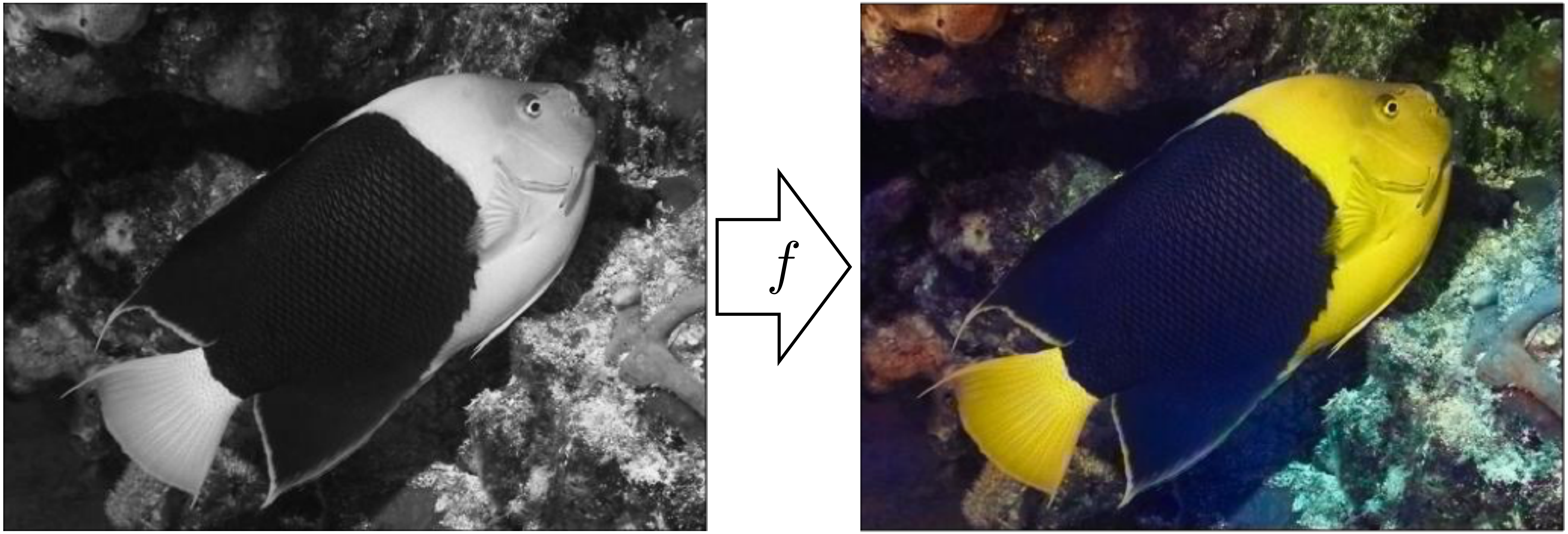
Grayscale image: **L channel**

$$\mathbf{x} \in \mathbb{R}^{H \times W \times 1}$$

Color information: **ab channels**

$$\mathbf{y} \in \mathbb{R}^{H \times W \times 2}$$

[Zhang et al., ECCV 2016]



Grayscale image: **L channel**

$$\mathbf{x} \in \mathbb{R}^{H \times W \times 1}$$

Color information: **ab channels**

$$\mathbf{y} \in \mathbb{R}^{H \times W \times 2}$$

[Zhang et al., ECCV 2016]

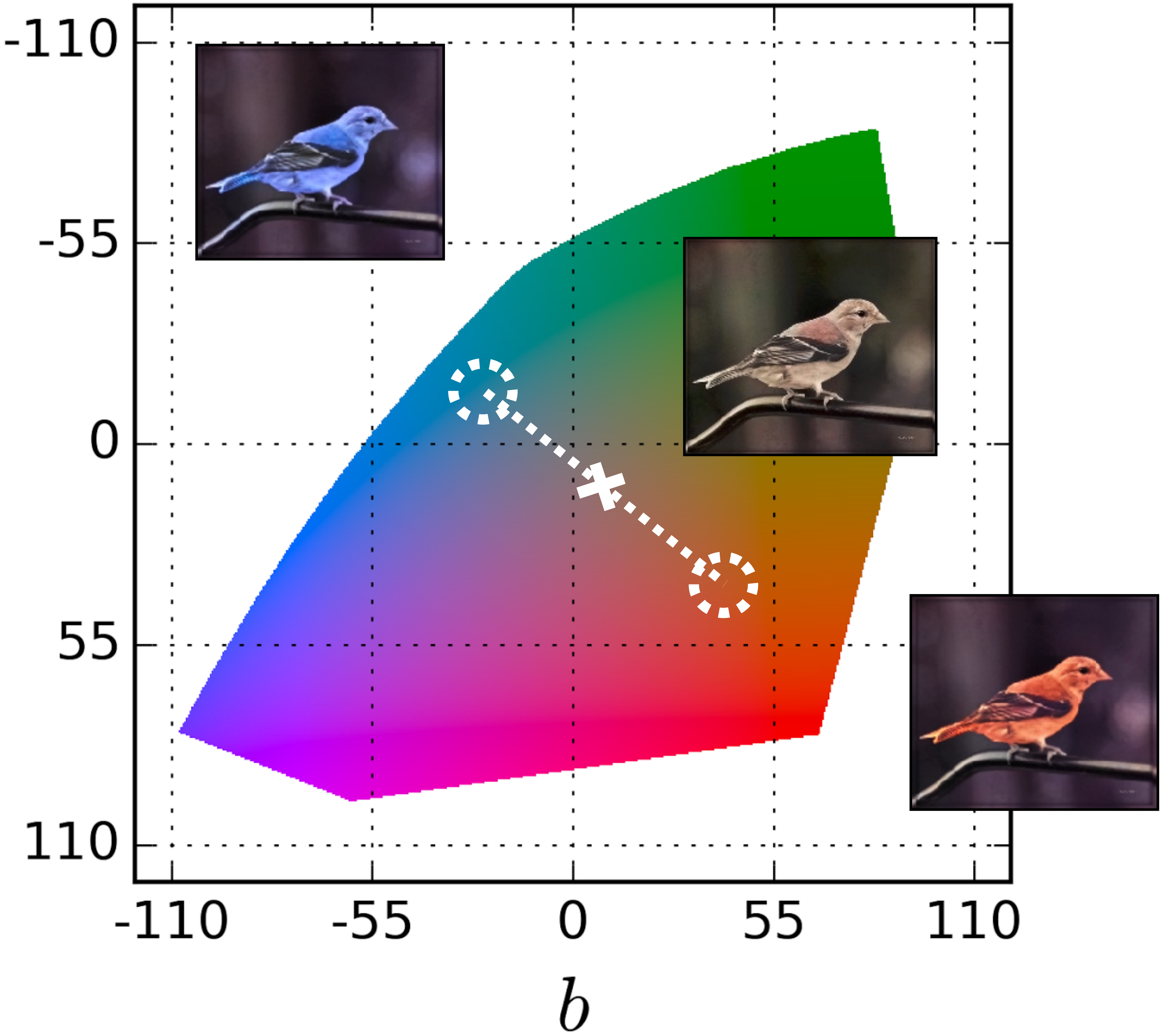
Choosing loss and representation

Input

Output

Ground truth

$$\mathcal{L}(f(\mathbf{x}), \mathbf{y}) = \|f(\mathbf{x}) - \mathbf{y}\|_2^2$$



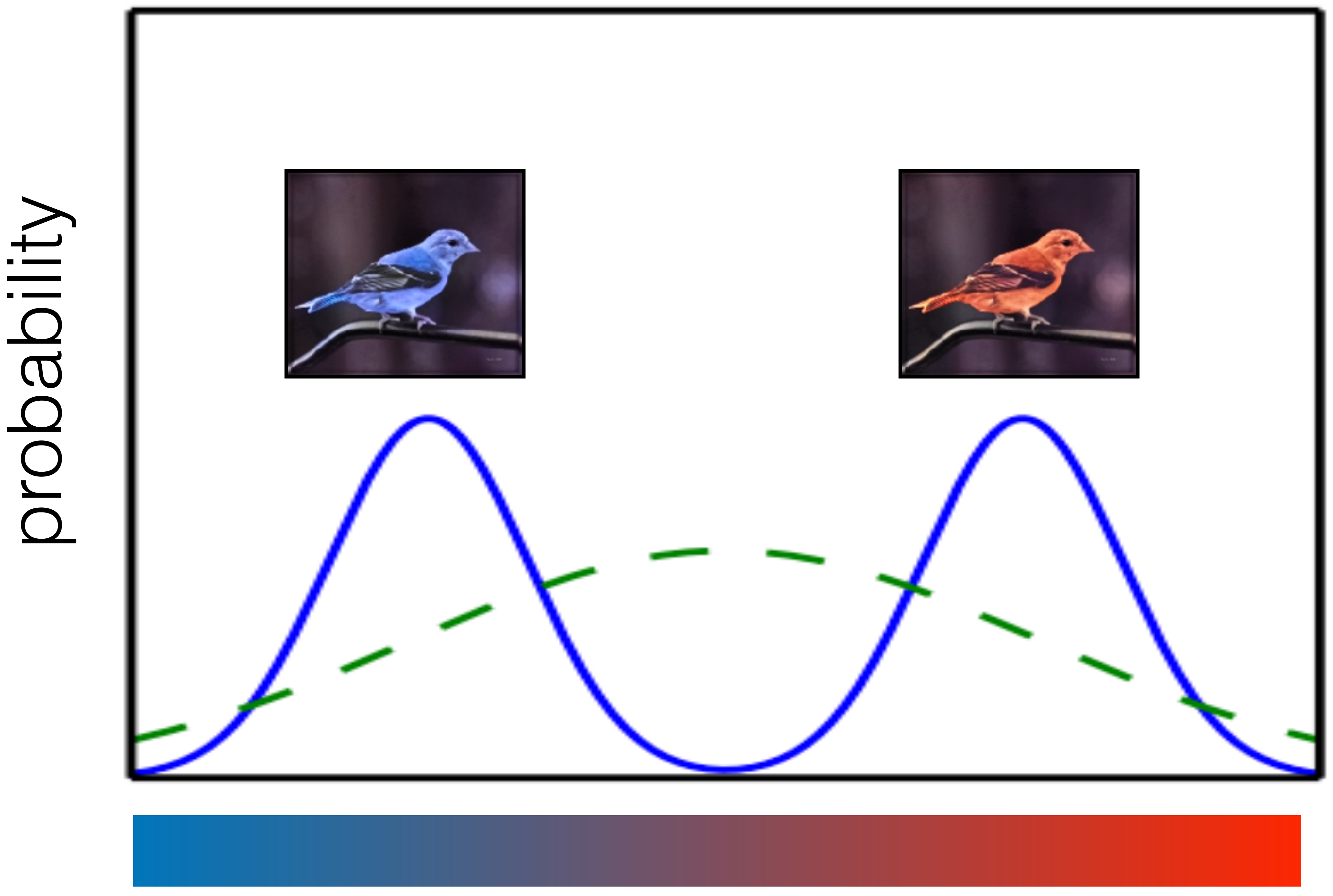
$$\mathcal{L}(f(\mathbf{x}),\mathbf{y}) = \|f(\mathbf{x}) - \mathbf{y}\|_2^2$$

Recall: least squares loss corresponds to the following modeling assumptions:

$$Y = f_\theta(X) + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma)$$

$$P_\theta(Y = y | X = x) \propto \exp \frac{-(y - f_\theta(x))^2}{2\sigma^2}$$

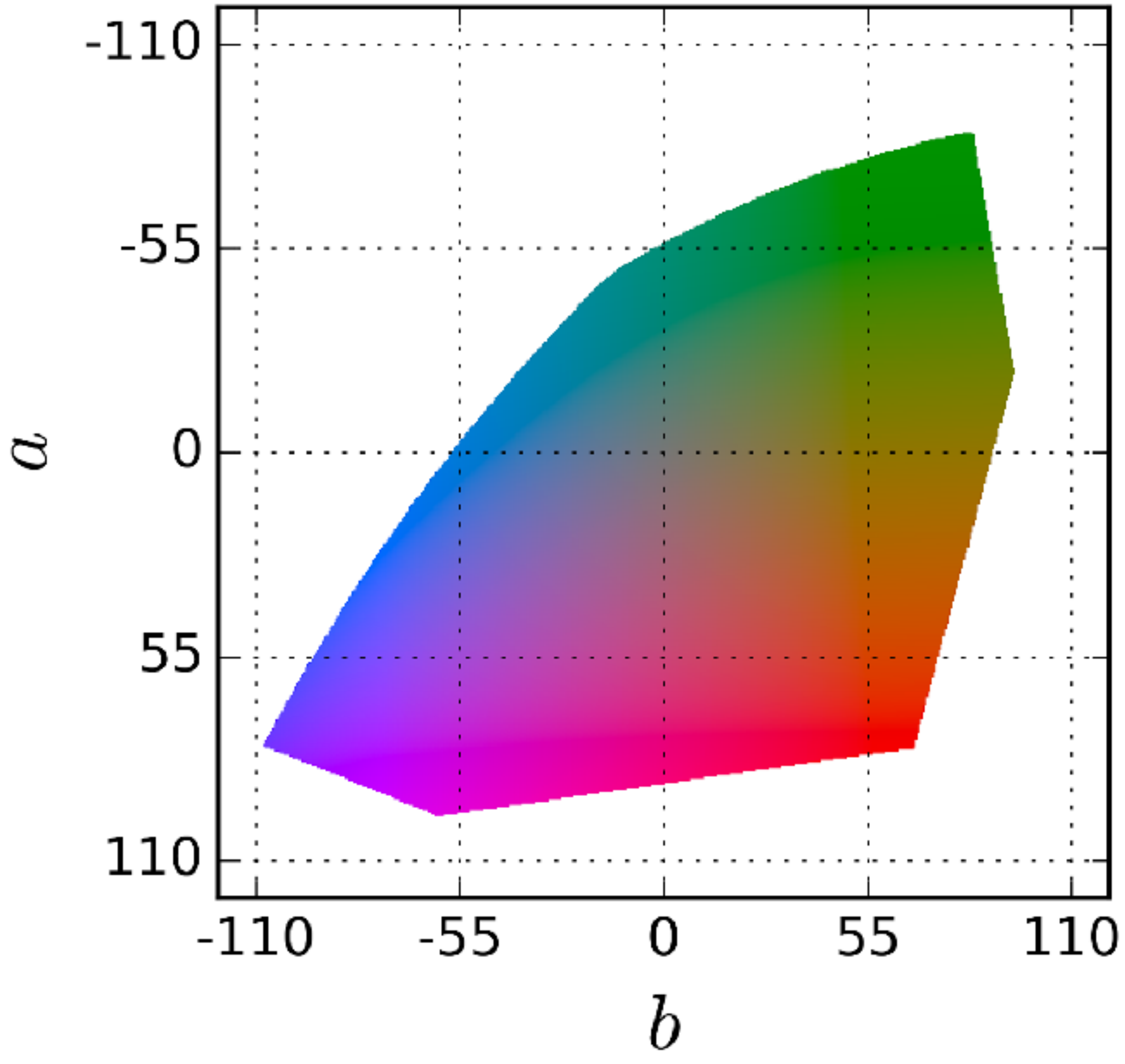
Prediction for a single pixel i,j



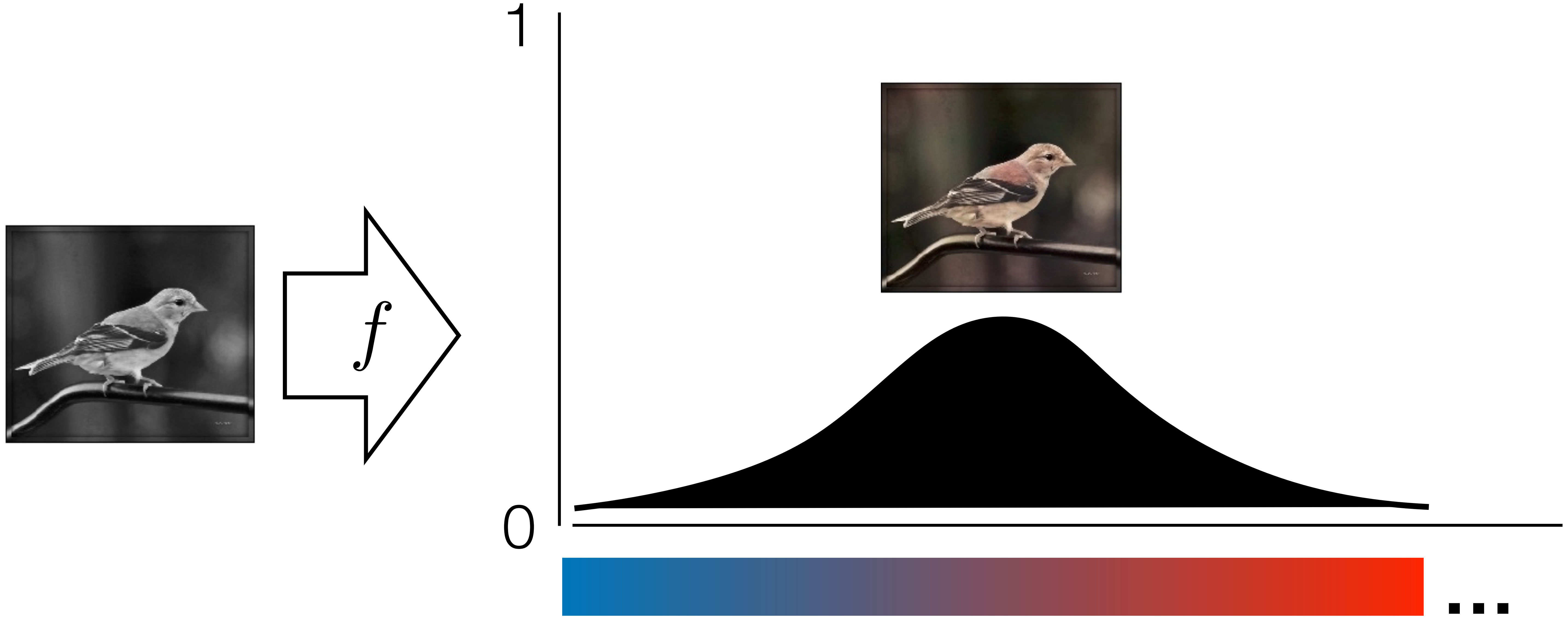
Best **Gaussian fit** to the **true data distribution** is to center the Gaussian on the mean of the data distribution.

fig modified from [Goodfellow, 2016]

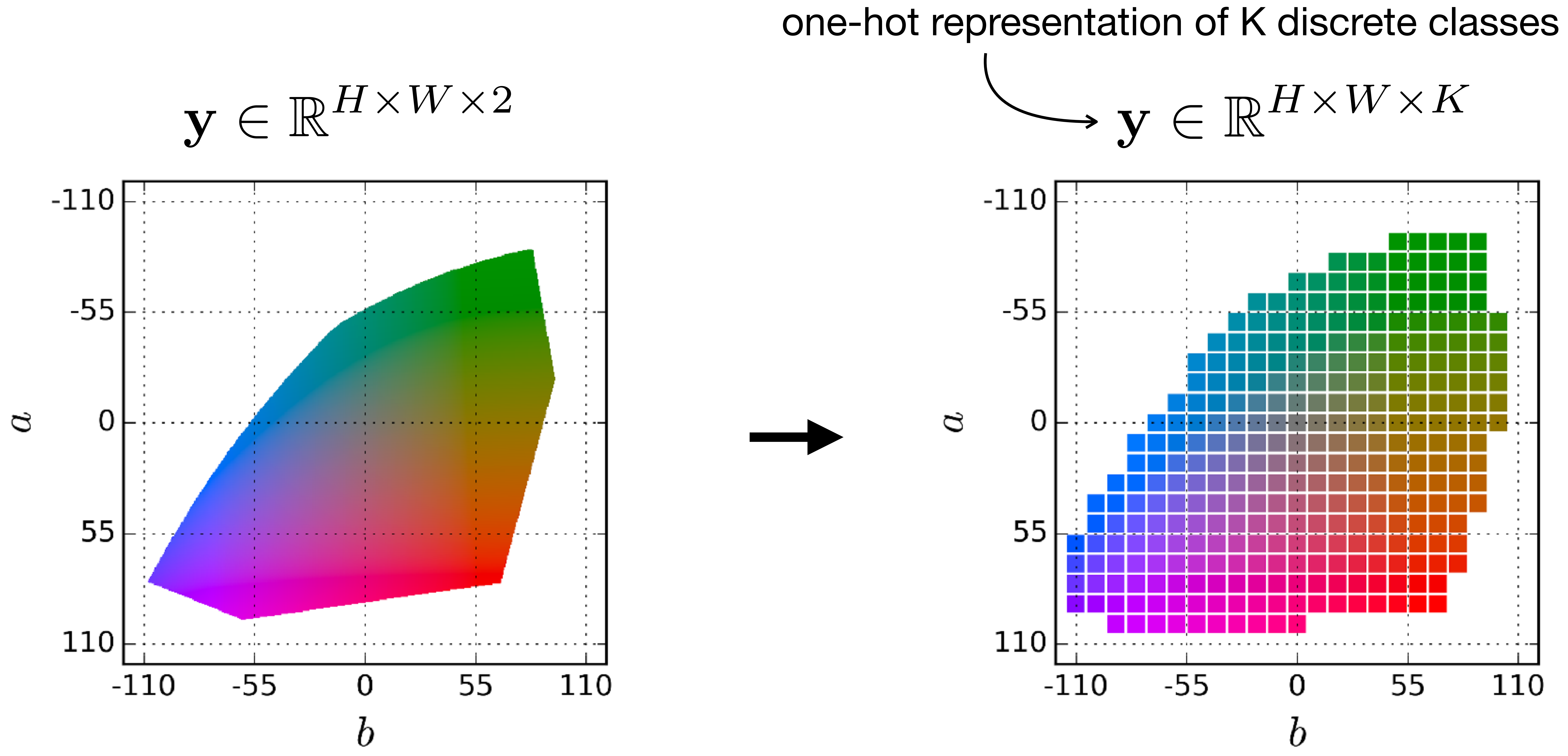
$$\mathbf{y} \in \mathbb{R}^{H \times W \times 2}$$



Prediction for a single pixel i,j

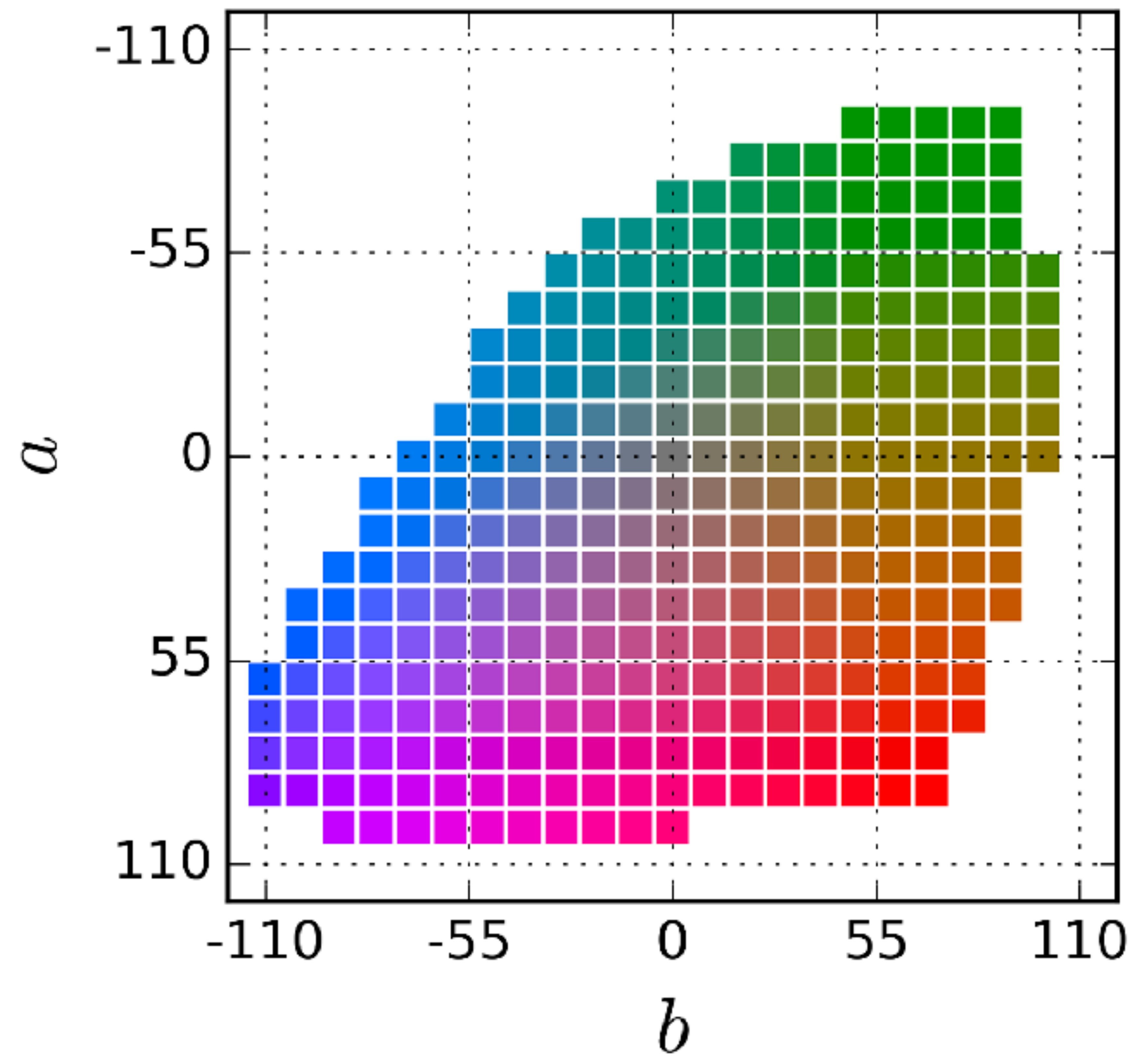


$$\mathcal{L}(f(\mathbf{x}), \mathbf{y}) = \|f(\mathbf{x}) - \mathbf{y}\|_2^2$$

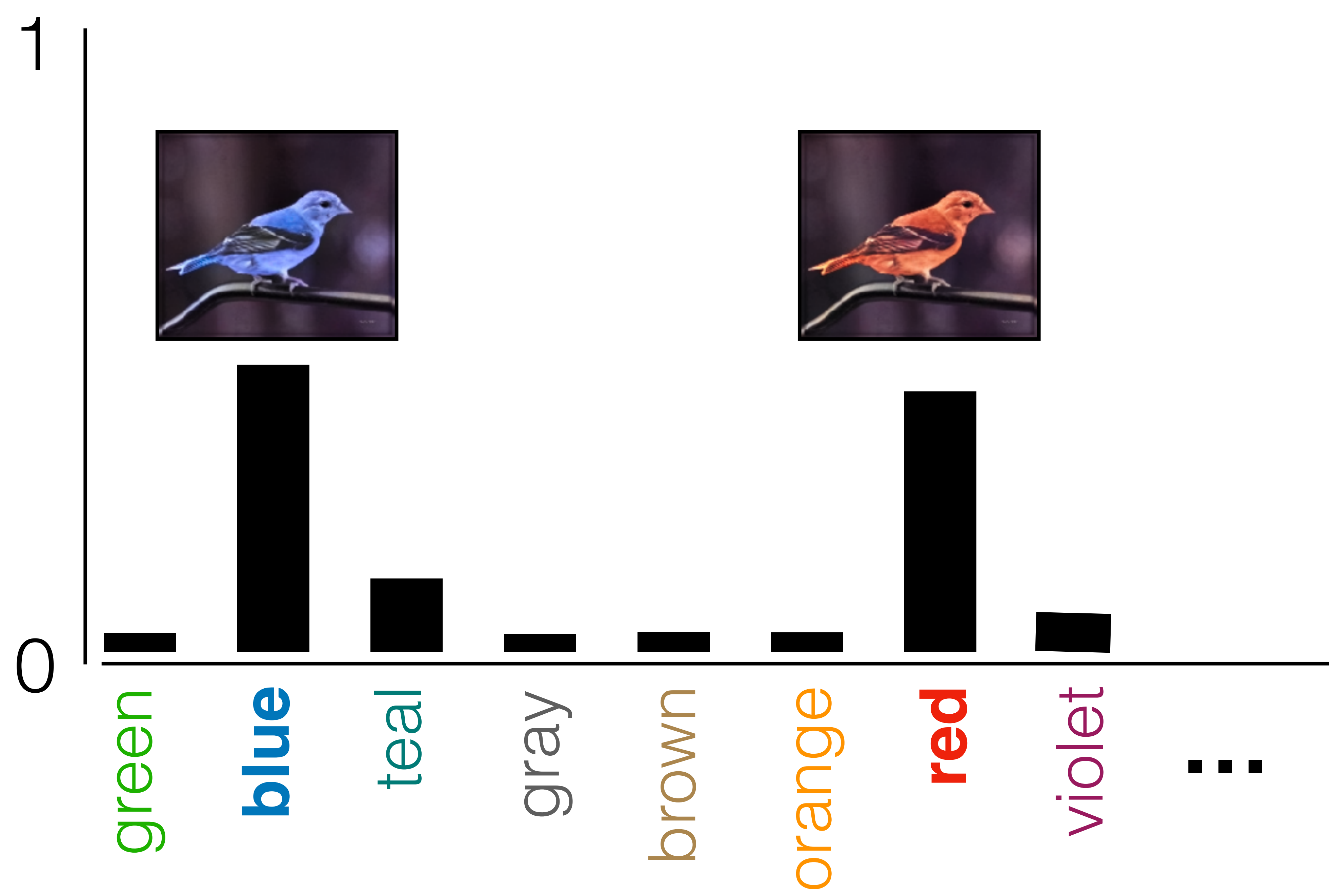


one-hot representation of K discrete classes
$\mathbf{y} \in \mathbb{R}^{H \times W \times 2}$
$\mathbf{y} \in \mathbb{R}^{H \times W \times K}$
-110
-55
0
55
110
a
b

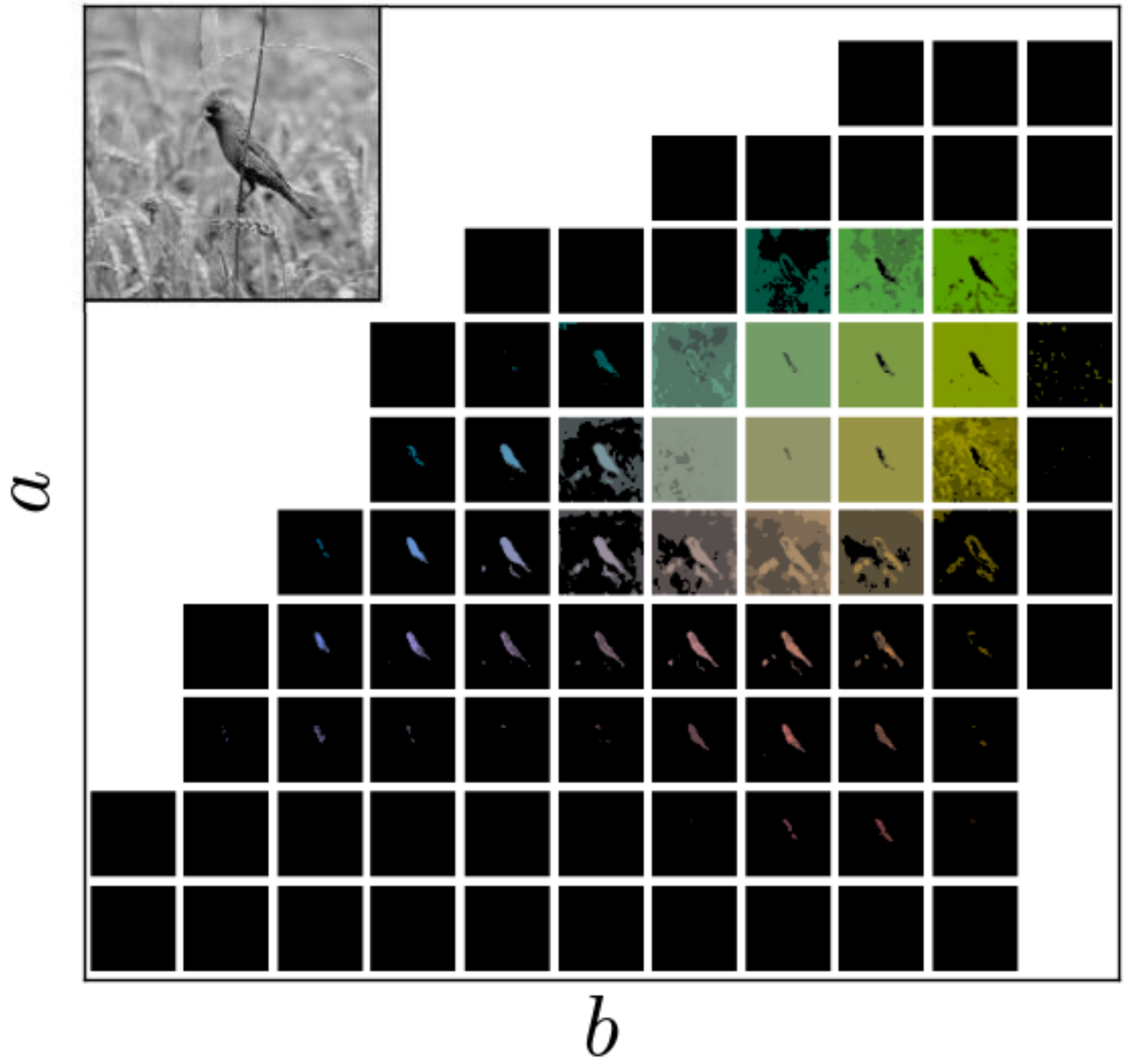
$$\mathbf{y} \in \mathbb{R}^{H \times W \times K}$$



Prediction for a single pixel i,j



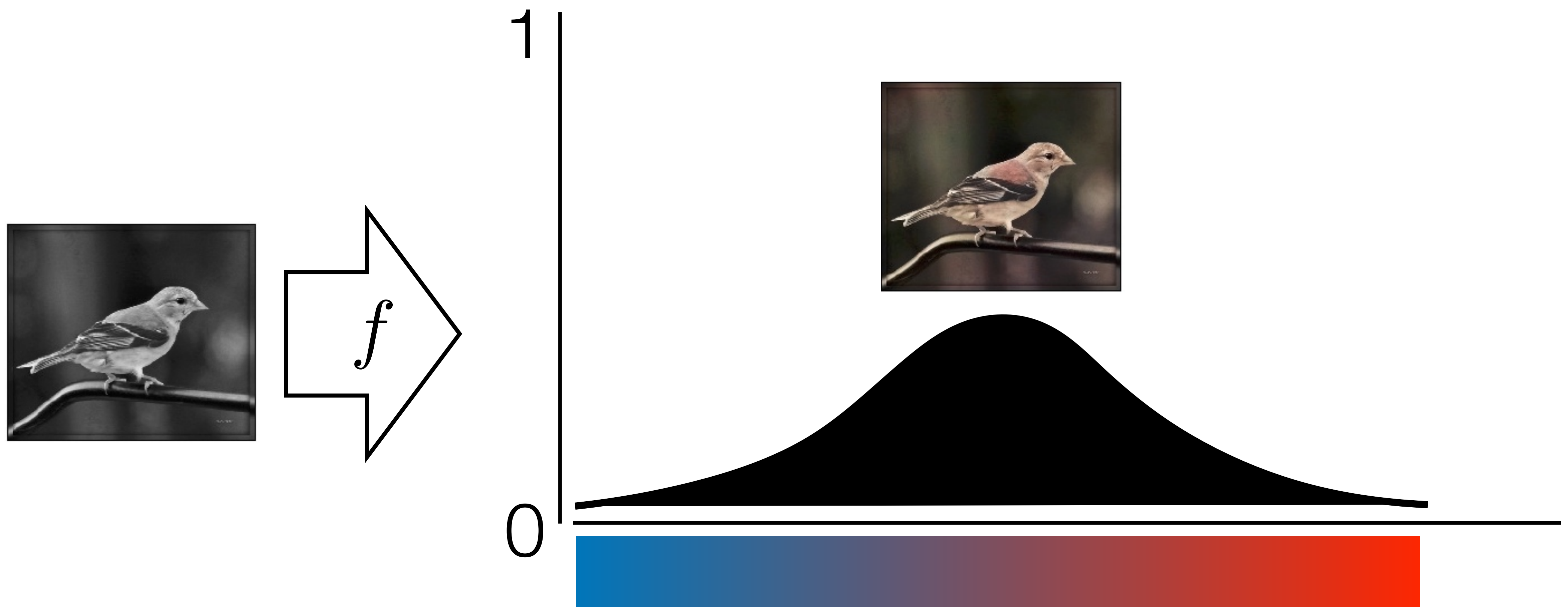
$$\mathcal{L}(\mathbf{y}, f_\theta(\mathbf{x})) = H(\mathbf{y}, \texttt{softmax}(f_\theta(\mathbf{x})))$$


a
b

Input | Zhang et al. 2016 | Ground truth

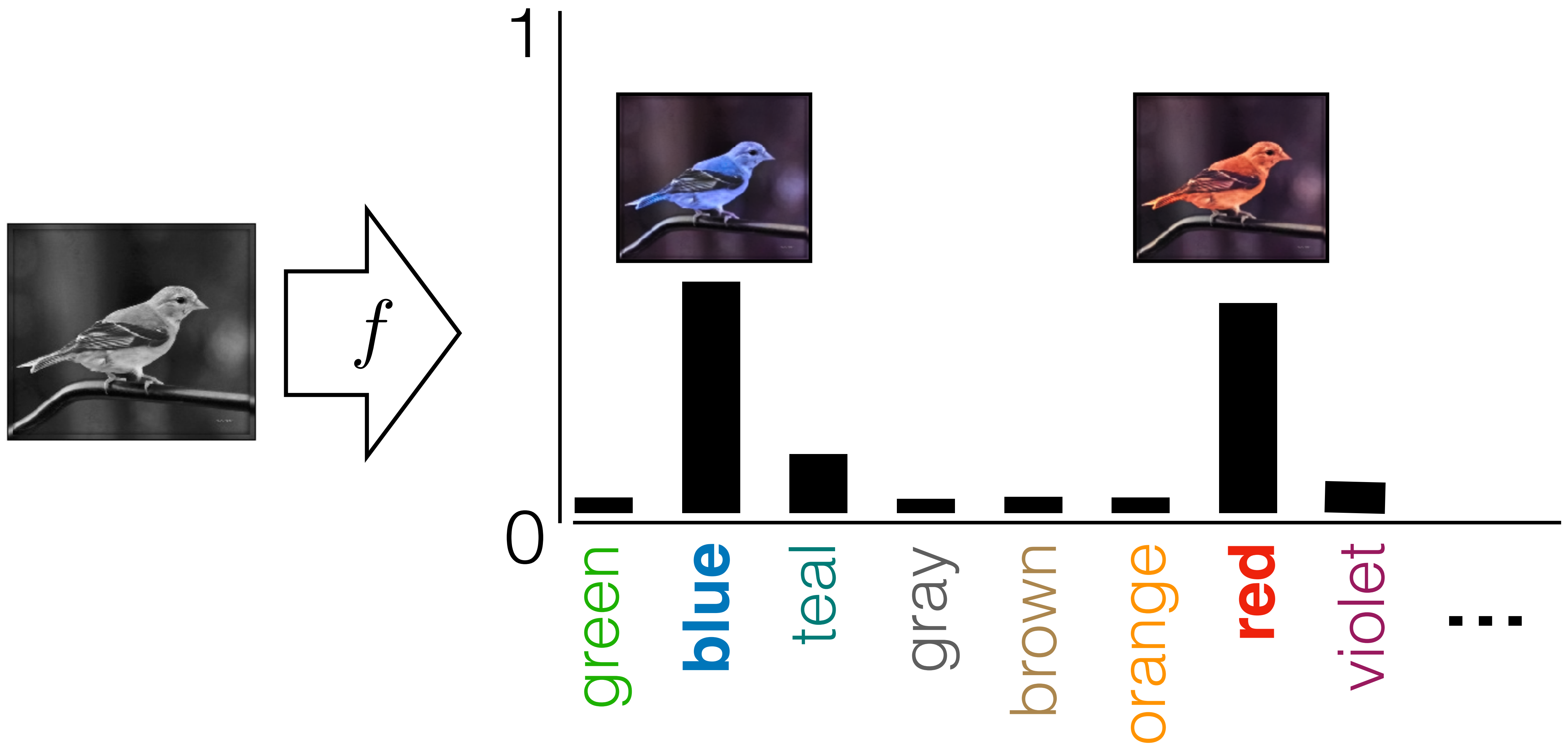
$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = H(\mathbf{y}, \texttt{softmax}(f_\theta(\mathbf{x})))$$

“Regression”



- Continuous-valued prediction
- (Usually) models unimodal distribution

“Classification”



- Discrete-valued prediction
- Models multimodal distribution

Instructive failure

Instructive failure

[from Reddit /u/SherySantucci]

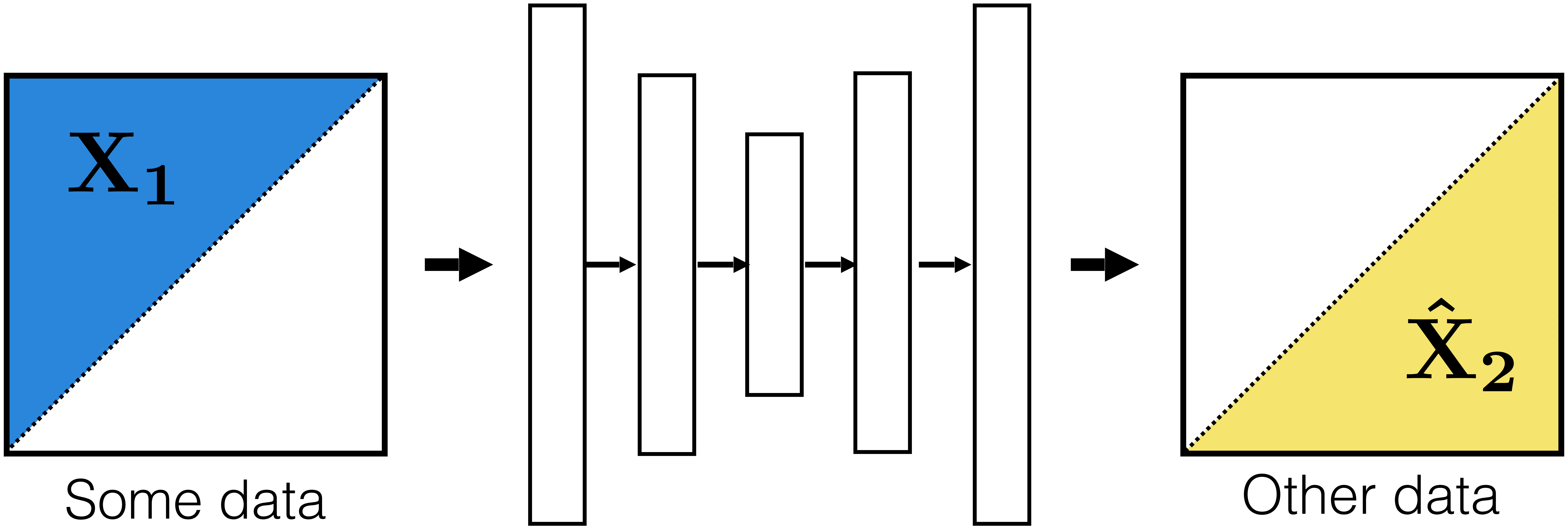
[Recolorized by Reddit ColorizeBot]

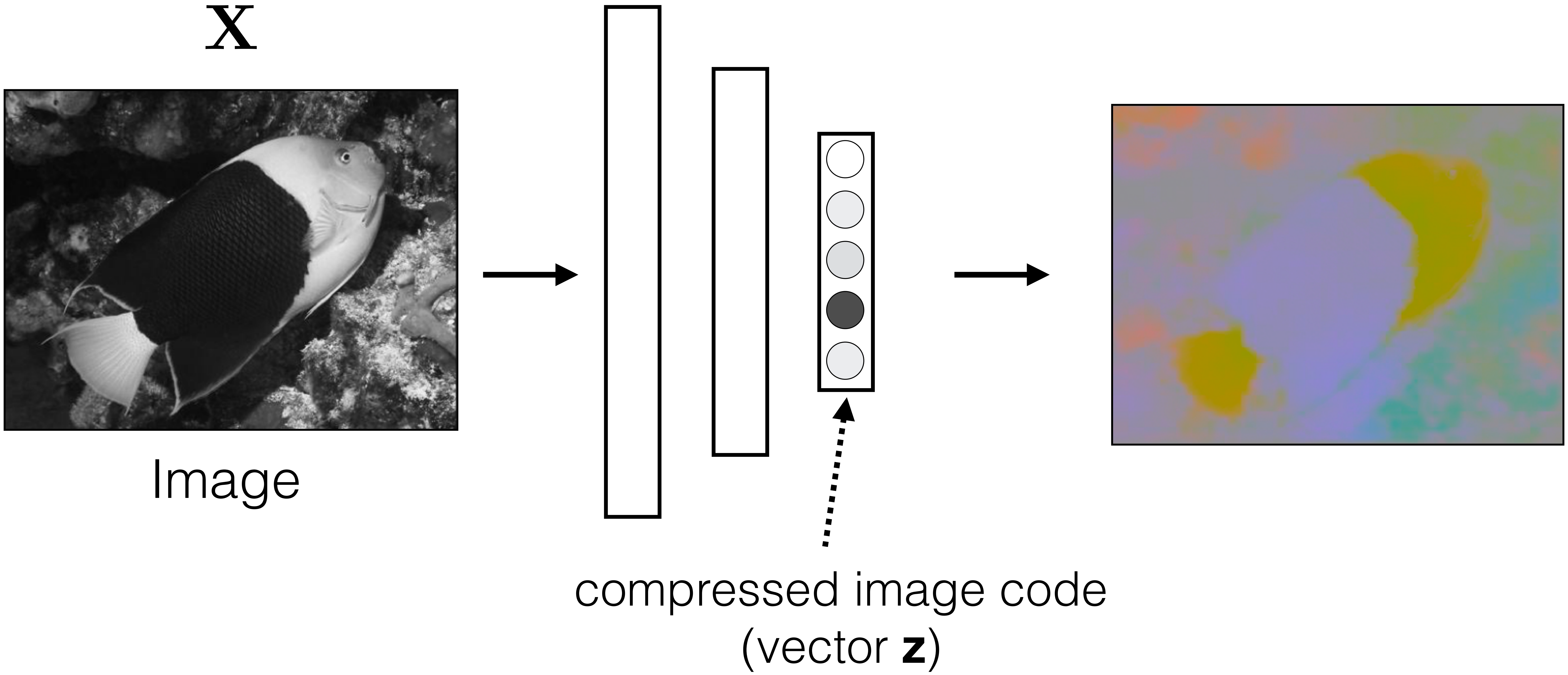
Photo taken by Reddit /u/ Timteroo,
Mural from street artist Eduardo Kobra

Recolorized by
Reddit
ColorizeBot

Data prediction

aka "self-supervised learning"

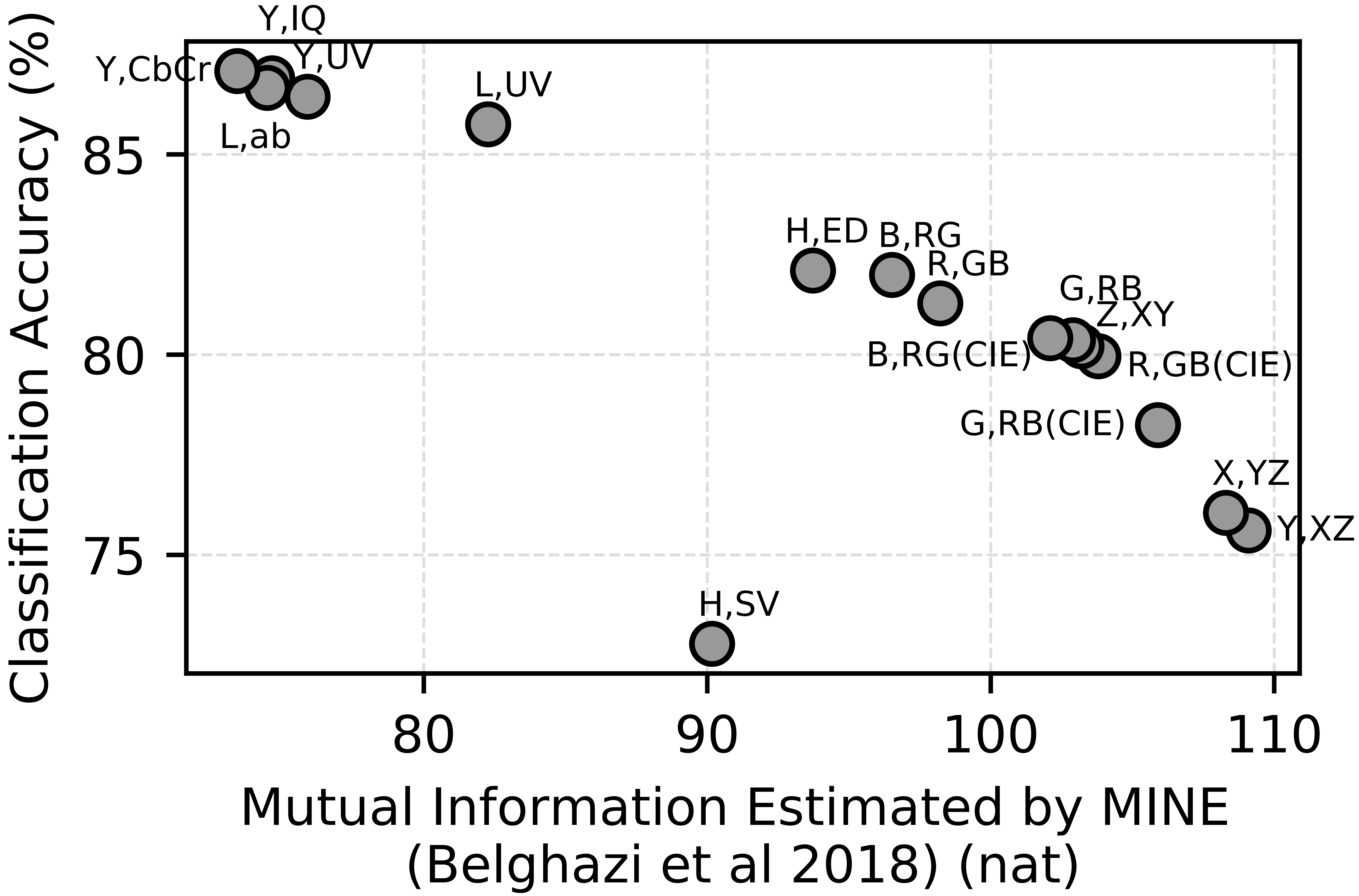




Is the code informative about object class y?

Logistic regression:

$$y = \sigma(\mathbf{W}\mathbf{z} + \mathbf{b})$$



Color space matters!

L—>ab much better than R —> GB

[“Contrastive Multiview Coding”, Tian, Krishnan, Isola, arXiv 2019]

Image colorization in a nutshell

Data

$\mathbf{x}$ $\mathbf{y}$

$\vdots$

$\mathbf{x} \in \mathbb{R}^{H \times W \times 1}$

$\mathbf{y} \in \mathbb{R}^{H \times W \times K}$

$\rightarrow$

Learner

Objective

$$\mathcal{L}(\mathbf{y}, f_\theta(\mathbf{x})) = H(\mathbf{y}, \texttt{softmax}(f_\theta(\mathbf{x})))$$

Hypothesis space

Convolutional neural net

Optimizer

Stochastic gradient descent

$\rightarrow f$