

# Linear Classifier and its Learning by Perceptron

Vojtěch Franc

February 24, 2020

Generic linear classifier

Instances of linear classifier

Perceptron algorithm

**XEP33SML – Structured Model Learning, Summer 2020**

## A generic linear classifier

- ◆  $\mathcal{X}$  is a set of observations and  $\mathcal{Y}$  is a finite set of hidden states
- ◆  $\phi: \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^n$  input-output feature map embedding  $\mathcal{X} \times \mathcal{Y}$  to  $\mathbb{R}^n$
- ◆ Generic linear classifier  $h: \mathcal{X} \rightarrow \mathcal{Y}$

$$h(x; \mathbf{w}) = \underset{y \in \mathcal{Y}(x)}{\text{Argmax}} \langle \mathbf{w}, \phi(x, y) \rangle$$

where  $\mathcal{Y}(x) \subseteq \mathcal{Y}$ .

- ◆ We will usually assume that  $\mathcal{Y}(x) = \mathcal{Y}, \forall x \in \mathcal{X}$ .

## Example: two-classes linear classifier

- ◆  $\mathcal{X}$  is a set of observations and  $\mathcal{Y} = \{+1, -1\}$  is a set of hidden labels
- ◆  $\phi: \mathcal{X} \rightarrow \mathbb{R}^d$  feature map embedding observations from  $\mathcal{X}$  to  $\mathbb{R}^n$
- ◆ Two-class linear classifier  $h: \mathcal{X} \rightarrow \mathcal{Y}$

$$h(x; \mathbf{w}, b) = \text{sign}(\langle \mathbf{w}, \phi(x) \rangle + b) = \begin{cases} +1 & \text{if } \langle \mathbf{w}, \phi(x) \rangle + b \geq 0 \\ -1 & \text{if } \langle \mathbf{w}, \phi(x) \rangle + b < 0 \end{cases}$$

- ◆ It is equivalent to

$$h(x; \mathbf{w}) = \underset{y \in \{+1, -1\}}{\text{Argmax}} y (\langle \mathbf{w}, \phi(x) \rangle + b) = \underset{y \in \{+1, -1\}}{\text{Argmax}} \langle \mathbf{w}', \phi(x, y) \rangle$$

for  $\phi(x, y) = [y \phi(x), y]$  and  $\mathbf{w}' = [\mathbf{w}, b]$ .

## Example: multi-class linear classifier

- ◆  $\mathcal{X}$  is a set of observations and  $\mathcal{Y} = \{1, \dots, Y\}$  is a set of class labels
- ◆ Multi-class linear classifier  $h: \mathcal{X} \rightarrow \mathcal{Y}$

$$h(x; \mathbf{w}) = \underset{y \in \mathcal{Y}}{\text{Argmax}} \langle \mathbf{w}_y, \phi(x) \rangle$$

where  $\phi: \mathcal{X} \rightarrow \mathbb{R}^d$  is a feature map  $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_Y) \in \mathbb{R}^{d \cdot Y}$  are parameters.

- ◆ We can write the score function as

$$\langle \mathbf{w}_y, \phi(x) \rangle = \langle \mathbf{w}, \phi(x, y) \rangle$$

where  $\phi: \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^{d \cdot Y}$  is

$$\phi(x, y) = (\mathbf{0}; \dots; \underbrace{\phi(x)}_{y\text{-th slot}}; \dots; \mathbf{0})$$

## Example: sequence classifier

- ◆  $\mathbf{x} = (x_1, \dots, x_L) \in \mathcal{X}^L$  sequence of  $L$  inputs
- ◆  $\mathbf{y} = (y_1, \dots, y_L) \in \mathcal{Y}^L$  sequence of  $L$  labels from  $\mathcal{Y} = \{A, \dots, Z\}$

For example:

$$\mathbf{x} = (x_1, x_2, x_3, x_4) \quad \mathbf{y} = (y_1, y_2, y_3, y_4)$$

JOHN	JOHN
BILL	BILL
⋮	⋮
DANA	DANA

## Example: sequence classifier

- ◆  $\mathbf{x} = (x_1, \dots, x_L) \in \mathcal{X}^L$  sequence of  $L$  images with characters
- ◆  $\mathbf{y} = (y_1, \dots, y_L) \in \mathcal{Y}^L$  sequence of  $L$  labels from  $\mathcal{Y} = \{A, \dots, Z\}$

For example:

$$JOHN = h(\text{JOHN}; \mathbf{w}) = \underset{\mathbf{y} \in \mathcal{Y}^L}{\text{Argmax}} \left\langle \phi(\text{JOHN}, \mathbf{y}), \mathbf{w} \right\rangle$$

$$\left\langle \phi(\text{JOHN}, AAAA), \mathbf{w} \right\rangle = 0.12$$

$$\left\langle \phi(\text{JOHN}, AAAB), \mathbf{w} \right\rangle = 0.10$$

⋮

$$\left\langle \phi(\text{JOHN}, JOHN), \mathbf{w} \right\rangle = 10.12$$

⋮

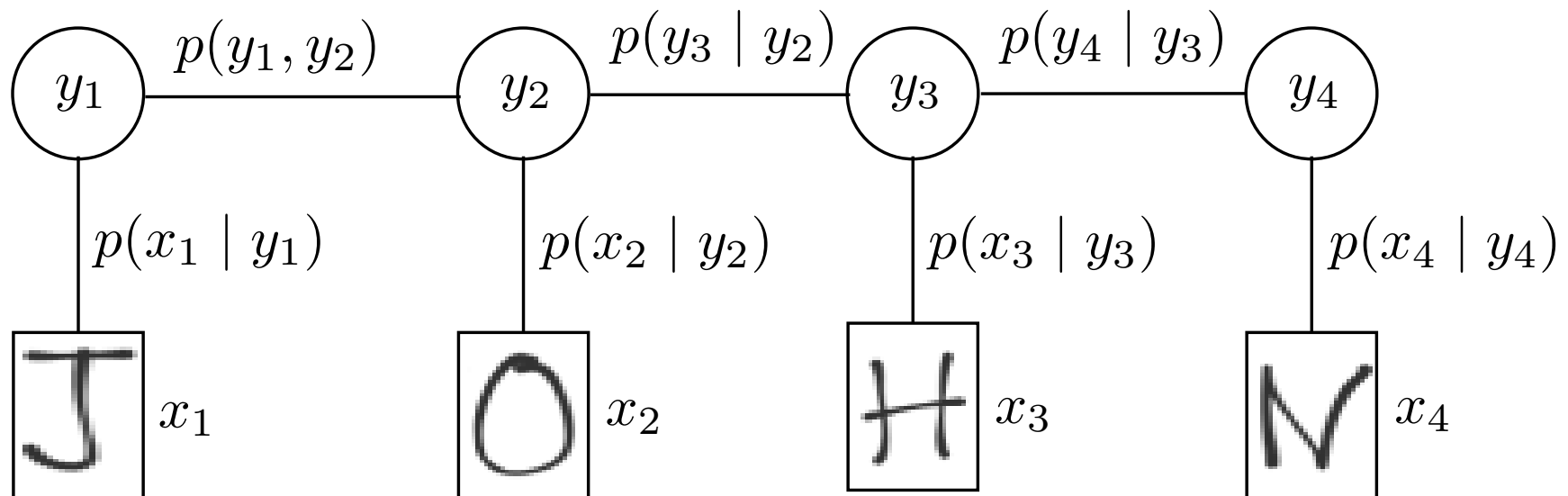
$$\left\langle \phi(\text{JOHN}, ZZZZ), \mathbf{w} \right\rangle = 0.34$$

## Example: sequence classifier

### Hidden Markov Chain model:

- ◆  $\mathbf{x} = (x_1, \dots, x_L) \in \mathcal{X}^L$  sequence of  $L$  inputs
- ◆  $\mathbf{y} = (y_1, \dots, y_L) \in \mathcal{Y}^L$  sequence of  $L$  labels from  $\mathcal{Y} = \{A, \dots, Z\}$
- ◆  $p(x_i | y_i)$  emission model
- ◆  $p(y_i | y_{i-1})$  transition model

$$p(\mathbf{x}, \mathbf{y}) = p(y_1) \prod_{i=2}^L p(y_i | y_{i-1}) \prod_{i=1}^L p(x_i | y_i)$$



## Example: sequence classifier

- ◆ The MAP estimate from HMC:

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}^L}{\text{Argmax}} \left( \log p(y_1) + \sum_{i=2}^L \log p(y_i | y_{i-1}) + \sum_{i=1}^L \log p(x_i | y_i) \right)$$



## Example: sequence classifier

- ◆ The MAP estimate from HMC:

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}^L}{\text{Argmax}} \left( \log p(y_1) + \sum_{i=2}^L \log p(y_i | y_{i-1}) + \sum_{i=1}^L \log p(x_i | y_i) \right)$$

- ◆ Let us assume the following parametrization:

$$\begin{aligned} \log p(y_1) &= \langle \mathbf{w}, \phi(y_1) \rangle \\ \log p(y_i | y_{i-1}) &= \langle \mathbf{w}, \phi(y_{i-1}, y_i) \rangle \\ \log p(x_i | y_i) &= \langle \mathbf{w}, \phi(x_i, y_i) \rangle \end{aligned}$$

- ◆ The MAP estimate becomes a linear classifier:

$$\hat{\mathbf{y}} = \underset{(y_1, \dots, y_L) \in \mathcal{Y}^L}{\text{Argmax}} \left\langle \mathbf{w}, \underbrace{\phi(y_1) + \sum_{i=2}^L \phi(y_{i-1}, y_i) + \sum_{i=1}^L \phi(x_i, y_i)}_{\phi(\mathbf{x}, \mathbf{y})} \right\rangle$$

## Example: Max-Sum (Markov-Network) classifier

### Setting:

- ◆  $(\mathcal{V}, \mathcal{E})$  is undirected graph;  $\mathcal{V}$  are parts and  $\mathcal{E} \subseteq \binom{|\mathcal{V}|}{2}$  are related parts
- ◆  $\mathbf{x} = (x_v \in \mathcal{X} \mid v \in \mathcal{V}) \in \mathcal{X}^{\mathcal{V}}$  inputs;  $\mathbf{y} = (y_v \in \mathcal{Y} \mid v \in \mathcal{V}) \in \mathcal{Y}^{\mathcal{V}}$  labels
- ◆  $q_v(x, y) = \langle \mathbf{w}, \phi_v(x, y) \rangle$
- ◆  $g_{vv'}(y, y') = \langle \mathbf{w}, \phi_{vv'}(y, y') \rangle$

**Linear Max-sum classifier:**  $h: \mathcal{X}^{\mathcal{V}} \rightarrow \mathcal{Y}^{\mathcal{V}}$  returns labeling

$$\begin{aligned}
 \hat{\mathbf{y}} &= \operatorname{Argmax}_{\mathbf{y} \in \mathcal{Y}^{\mathcal{V}}} \left( \sum_{v \in \mathcal{V}} g_v(x_v, y_v) + \sum_{(v, v') \in \mathcal{E}} g_{vv'}(y_v, y_{v'}) \right) \\
 &= \operatorname{Argmax}_{\mathbf{y} \in \mathcal{Y}^{\mathcal{V}}} \left\langle \mathbf{w}, \underbrace{\sum_{v \in \mathcal{V}} \phi(x_v, y_v) + \sum_{(v, v') \in \mathcal{E}} \phi(y_v, y_{v'})}_{\phi(\mathbf{x}, \mathbf{y})} \right\rangle
 \end{aligned}$$

## Learning by Empirical Risk Minimization

- ◆  $\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty)$  loss function; we assume  $\ell(y, y') = 0$  iff  $y = y'$ .
- ◆ Find parameters  $\mathbf{w}$  of  $h(x; \mathbf{w})$  which minimize the expected risk

$$R(\mathbf{w}) = \mathbb{E}_{(x,y) \sim p} \left( \ell(y, h(x; \mathbf{w})) \right)$$

## Learning by Empirical Risk Minimization

- ◆  $\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty)$  loss function; we assume  $\ell(y, y') = 0$  iff  $y = y'$ .
- ◆ Find parameters  $\mathbf{w}$  of  $h(x; \mathbf{w})$  which minimize the expected risk

$$R(\mathbf{w}) = \mathbb{E}_{(x,y) \sim p} \left( \ell(y, h(x; \mathbf{w})) \right)$$

- ◆ The Empirical Risk Minimization principle leads to solving

$$\mathbf{w}^* \in \underset{\mathbf{w} \in \mathbb{R}^n}{\text{Argmin}} R_{\mathcal{T}^m}(\mathbf{w})$$

where the empirical risk is

$$R_{\mathcal{T}^m}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \ell(y^i, h(x^i; \mathbf{w}))$$

and  $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \dots, m\}$  are training examples drawn from i.i.d. with distribution  $p(x, y)$ .

## Learning linear classifier from separable examples

- ◆ A correctly classified example  $(x^i, y^i)$ , that is,

$$y^i = h(x^i; \mathbf{w}) = \underset{y \in \mathcal{Y}}{\text{Argmax}} \langle \mathbf{w}, \phi(x^i, y) \rangle$$

implies

$$\langle \phi(x^i, y^i), \mathbf{w} \rangle > \langle \phi(x^i, y), \mathbf{w} \rangle, \quad \forall y \in \mathcal{Y} \setminus \{y^i\}$$

**Definition 1.** The examples  $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \dots, m\}$  are linearly separable w.r.t. joint feature map  $\phi: \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^n$  if there exists  $\mathbf{w} \in \mathbb{R}^n$  such that

$$\langle \phi(x^i, y^i), \mathbf{w} \rangle > \langle \phi(x^i, y), \mathbf{w} \rangle, \quad \forall i \in \{1, \dots, m\}, y \in \mathcal{Y} \setminus \{y^i\}$$

## Example: sequence classifier

$$\mathcal{T}^m = \{(\text{JOHN}, \text{JOHN}), (\text{BILL}, \text{BILL}), \dots\}$$

$$\left. \begin{array}{l} \langle \phi(\text{JOHN}, \text{JOHN}), \mathbf{w} \rangle > \langle \phi(\text{JOHN}, \text{AAAA}), \mathbf{w} \rangle \\ \langle \phi(\text{JOHN}, \text{JOHN}), \mathbf{w} \rangle > \langle \phi(\text{JOHN}, \text{AAAB}), \mathbf{w} \rangle \\ \vdots \\ \langle \phi(\text{JOHN}, \text{JOHN}), \mathbf{w} \rangle > \langle \phi(\text{JOHN}, \text{ZZZZ}), \mathbf{w} \rangle \end{array} \right\} \begin{array}{l} 26^4 - 1 \\ \text{inequalities} \end{array}$$

## Example: sequence classifier

$$\mathcal{T}^m = \{(\text{JOHN}, \text{JOHN}), (\text{BILL}, \text{BILL}), \dots\}$$

$$\left. \begin{array}{l} \langle \phi(\text{JOHN}, \text{JOHN}), \mathbf{w} \rangle > \langle \phi(\text{JOHN}, \text{AAAA}), \mathbf{w} \rangle \\ \langle \phi(\text{JOHN}, \text{JOHN}), \mathbf{w} \rangle > \langle \phi(\text{JOHN}, \text{AAAB}), \mathbf{w} \rangle \\ \vdots \\ \langle \phi(\text{JOHN}, \text{JOHN}), \mathbf{w} \rangle > \langle \phi(\text{JOHN}, \text{ZZZZ}), \mathbf{w} \rangle \end{array} \right\} \begin{array}{l} 26^4 - 1 \\ \text{inequalities} \end{array}$$

$$\left. \begin{array}{l} \langle \phi(\text{BILL}, \text{BILL}), \mathbf{w} \rangle > \langle \phi(\text{BILL}, \text{AAAA}), \mathbf{w} \rangle \\ \langle \phi(\text{BILL}, \text{BILL}), \mathbf{w} \rangle > \langle \phi(\text{BILL}, \text{AAAB}), \mathbf{w} \rangle \\ \vdots \\ \langle \phi(\text{BILL}, \text{BILL}), \mathbf{w} \rangle > \langle \phi(\text{JOHN}, \text{ZZZZ}), \mathbf{w} \rangle \end{array} \right\} \begin{array}{l} 26^4 - 1 \\ \text{inequalities} \end{array}$$

## Example: noise-free setting

- ◆  $x \in \mathcal{X}$  is randomly generated according to some  $p(x)$
- ◆  $y \in \mathcal{Y}$  are labels ( $\mathcal{Y}$  is finite) generated from

$$p(y | x) = [h^*(x) = y]$$

where  $h^* : \mathcal{X} \rightarrow \mathcal{Y}$  is a function.

- ◆ Under assumption that  $h^*(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \phi(x, y) \rangle$  the examples

$$\mathcal{T}^m = \{(x^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \dots, m\}$$

generated from  $p(x, y) = p(x) p(y | x)$  are linearly separable.



## (Generic) Perceptron algorithm

- ◆ **Task:** given a set of points  $\{\mathbf{a}^i \in \mathbb{R}^n \mid i = 1, 2, \dots, l\}$  we want to find  $\mathbf{w} \in \mathbb{R}^n$  such that

$$\langle \mathbf{w}, \mathbf{a}^i \rangle > 0, \quad \forall i \in \{1, 2, \dots, l\} \quad (1)$$

- ◆ **Perceptron:**

1.  $\mathbf{w} \leftarrow \mathbf{0}$
2. Find a violating  $\langle \mathbf{w}, \mathbf{a}^i \rangle \leq 0, i \in \{1, 2, \dots, l\}$
3. If there is no violating inequality return  $\mathbf{w}$  otherwise update

$$\mathbf{w} \leftarrow \mathbf{w} + \mathbf{a}^i$$

and go to step 2.

## Convergence of Perceptron

**Theorem 1.** For any linearly separable points  $\{\mathbf{a}^i \in \mathbb{R}^n \mid i = 1, 2, \dots, l\}$ , the Perceptron algorithm terminates in

$$\frac{A^2}{\gamma^2}$$

steps at most where

$$A = \max_{i=1, \dots, l} \|\mathbf{a}^i\|_2 \quad \text{and} \quad \gamma = \max_{\|\mathbf{w}\|=1} \min_{i=1, \dots, l} \frac{\langle \mathbf{w}, \mathbf{a}^i \rangle}{\|\mathbf{w}\|_2}$$

- ◆ Note that the upper bound  $\frac{A^2}{\gamma^2}$  does not depend on the number of points  $l$ .

## Structured Output Perceptron

- ◆ Learning  $h(x; \mathbf{w}) = \text{Argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \phi(x, y) \rangle$  from examples  $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \dots, m\}$  leads to solving

$$\langle \phi(x^i, y^i) - \phi(x^i, y), \mathbf{w} \rangle > 0, \quad \forall i \in \{1, \dots, m\}, y \in \mathcal{Y} \setminus \{y^i\}$$

## Structured Output Perceptron

- ◆ Learning  $h(x; \mathbf{w}) = \text{Argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \phi(x, y) \rangle$  from examples  $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \dots, m\}$  leads to solving

$$\langle \phi(x^i, y^i) - \phi(x^i, y), \mathbf{w} \rangle > 0, \quad \forall i \in \{1, \dots, m\}, y \in \mathcal{Y} \setminus \{y^i\}$$

- ◆ **Algorithm:**

1.  $\mathbf{w} \leftarrow \mathbf{0}$
2. Find a misclassified example  $(x^i, y^i) \in \mathcal{T}^m$  such that

$$y^i \neq \hat{y}^i = \text{Argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \phi(x^i, y) \rangle \quad \text{prediction problem}$$

3. If there is no misclassified example return  $\mathbf{w}$  otherwise update

$$\mathbf{w} \leftarrow \mathbf{w} + \phi(x^i, y^i) - \phi(x^i, \hat{y}^i) \quad \text{parameter update}$$

and go to step 2.

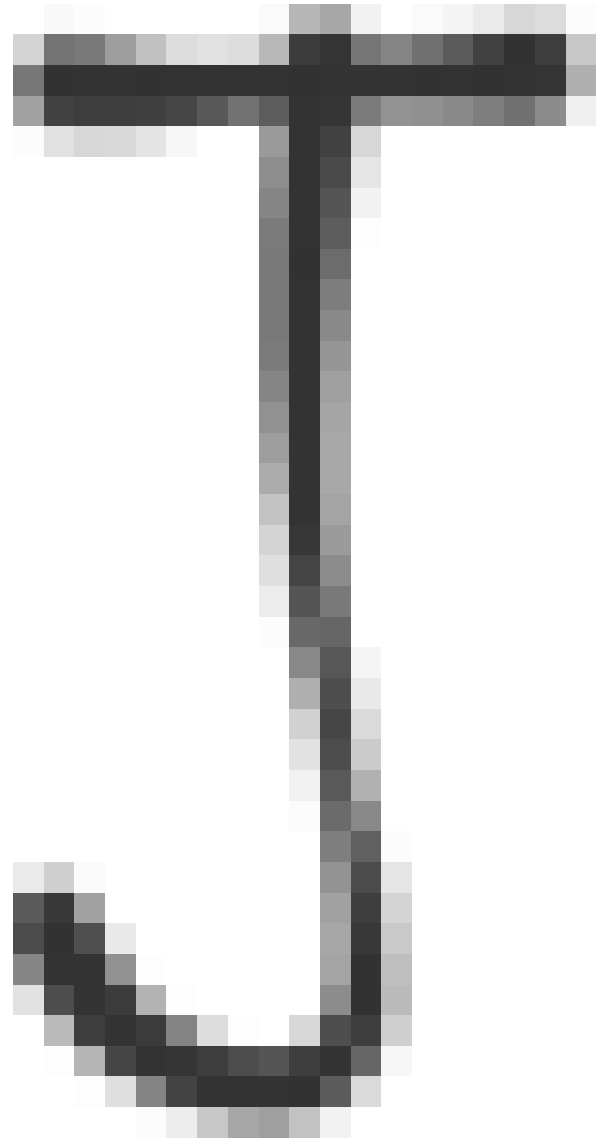
## Convergence of Structured Output Perceptron

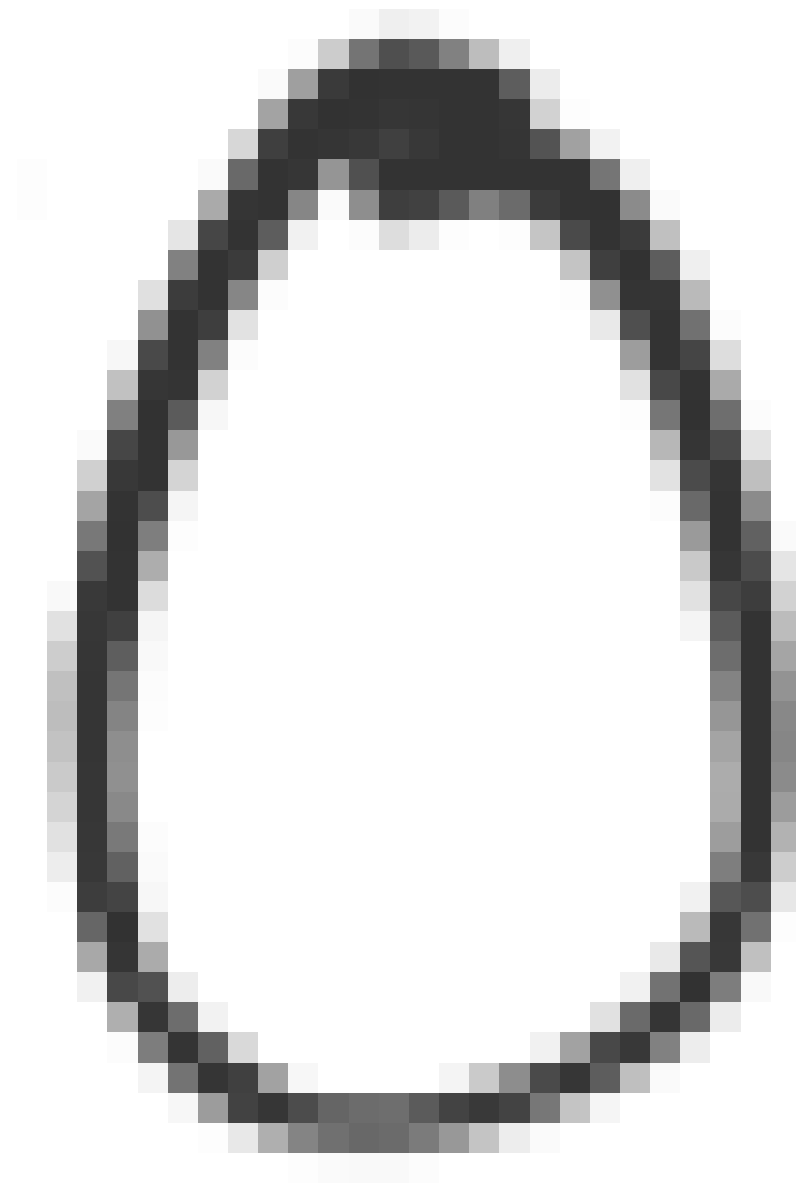
- ◆ By Theorem 1 we have a guarantee that for linearly separable training set  $\{(x^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, 2, \dots, m\}$  the SO-Perceptron terminates after at most  $\frac{A^2}{\gamma^2}$  iterations where

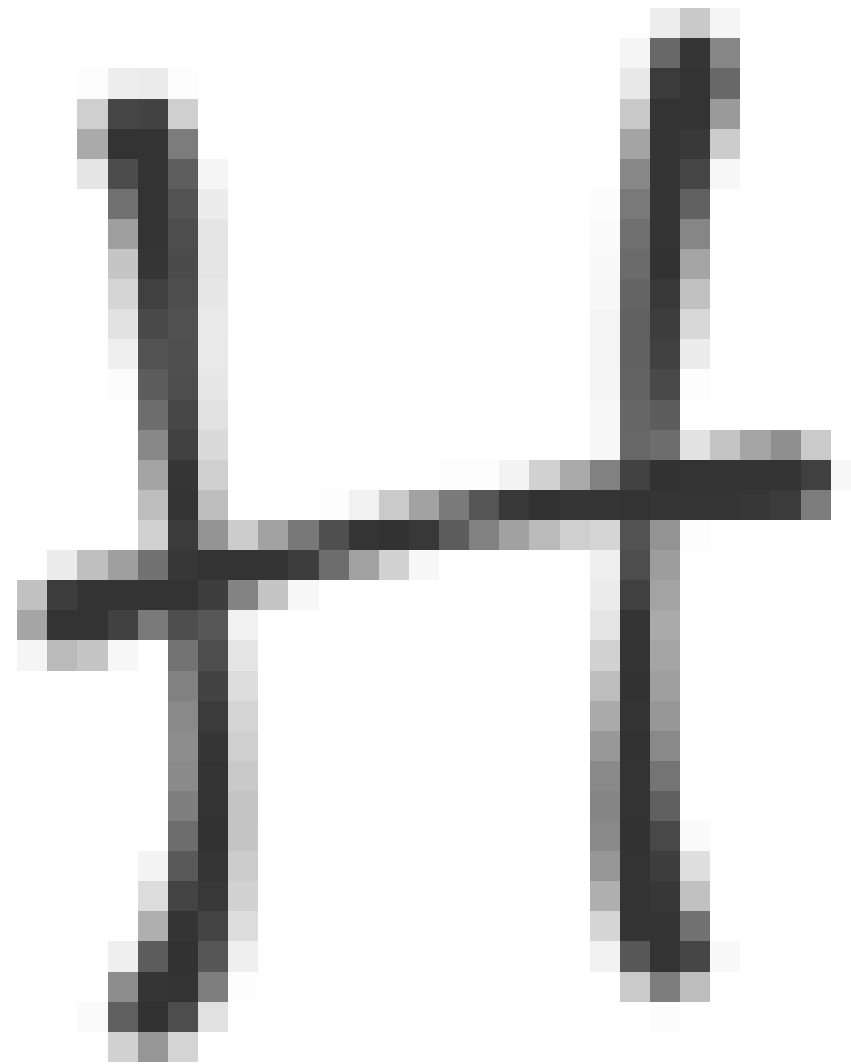
$$A = \max_{\substack{i=1,2,\dots,m \\ y \in \mathcal{Y} \setminus \{y^i\}}} \|\phi(x^i, y^i) - \phi(x^i, y)\| \leq 2 \max_{x \in \mathcal{X}, y \in \mathcal{Y}} \|\phi(x, y)\|$$

and

$$\gamma = \max_{\|\mathbf{w}\|=1} \min_{\substack{i=1,2,\dots,m \\ y \in \mathcal{Y} \setminus \{y^i\}}} \frac{\langle \mathbf{w}, \phi(x^i, y^i) - \phi(x^i, y) \rangle}{\|\mathbf{w}\|_2}$$



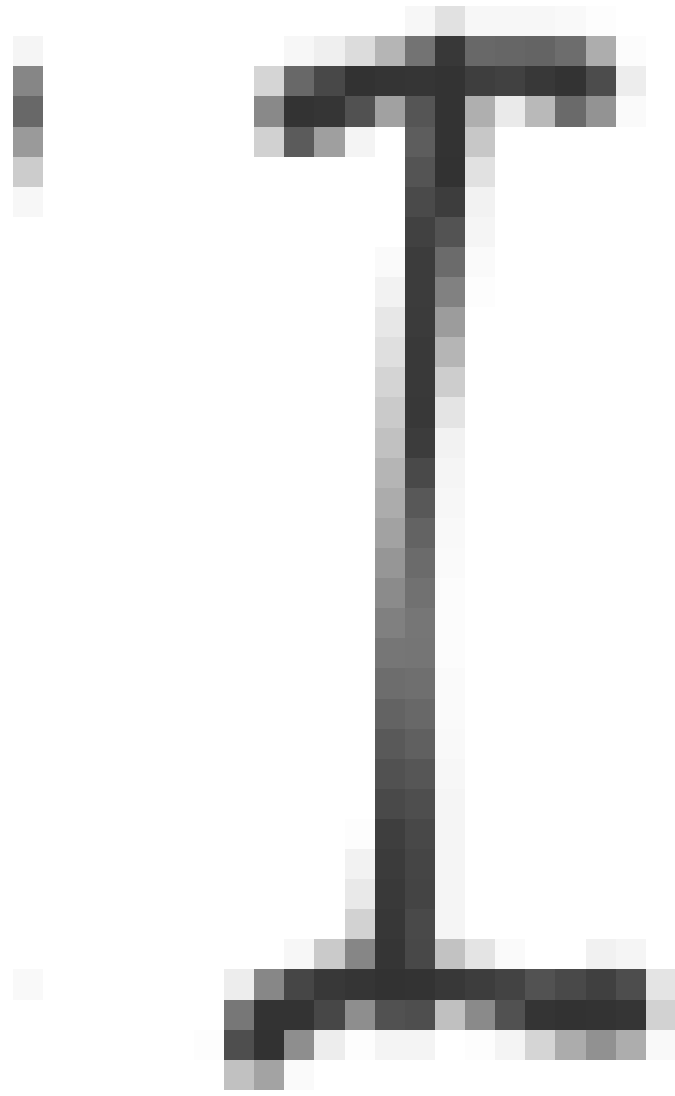


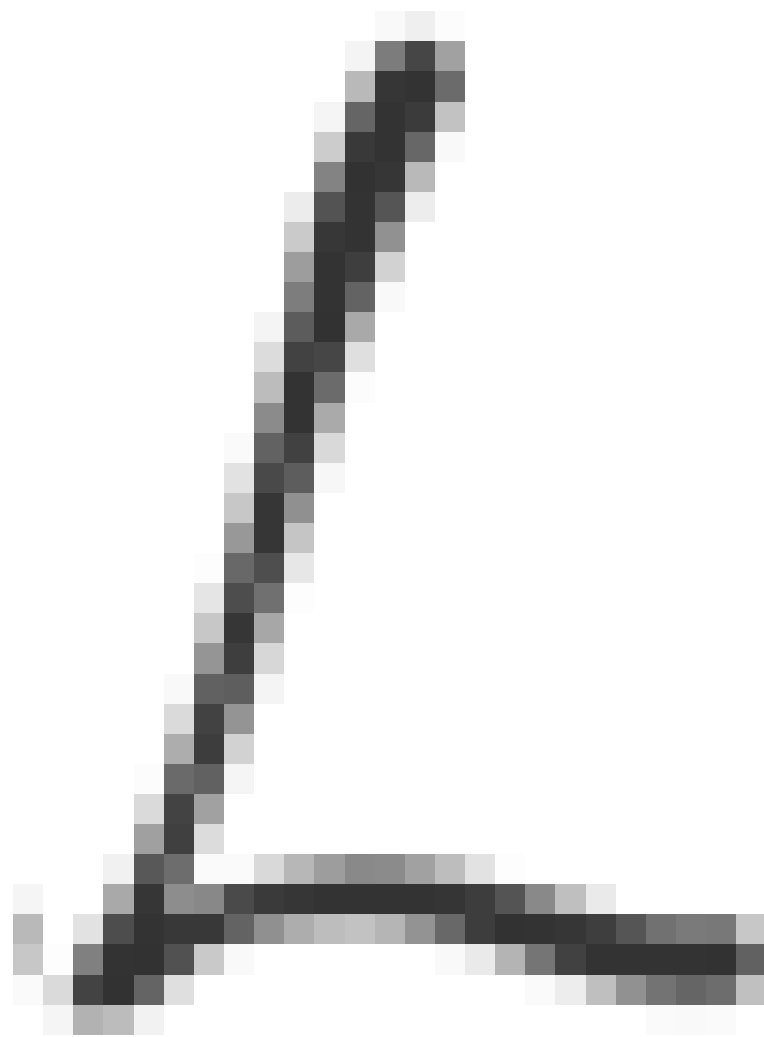


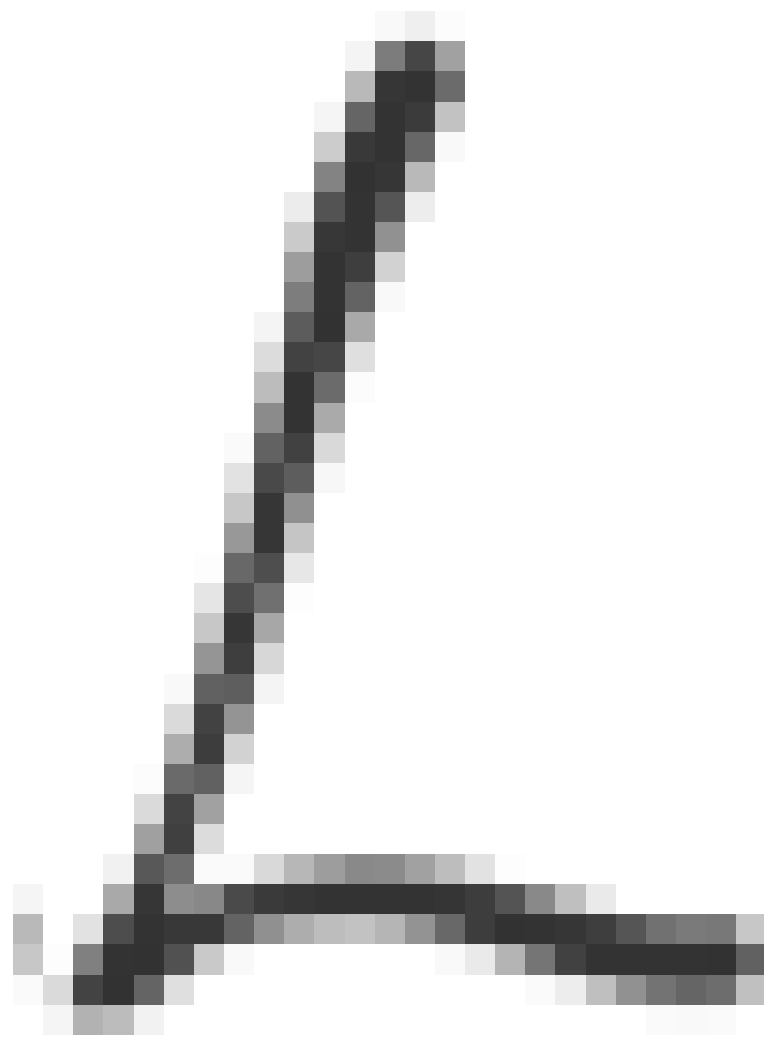


W

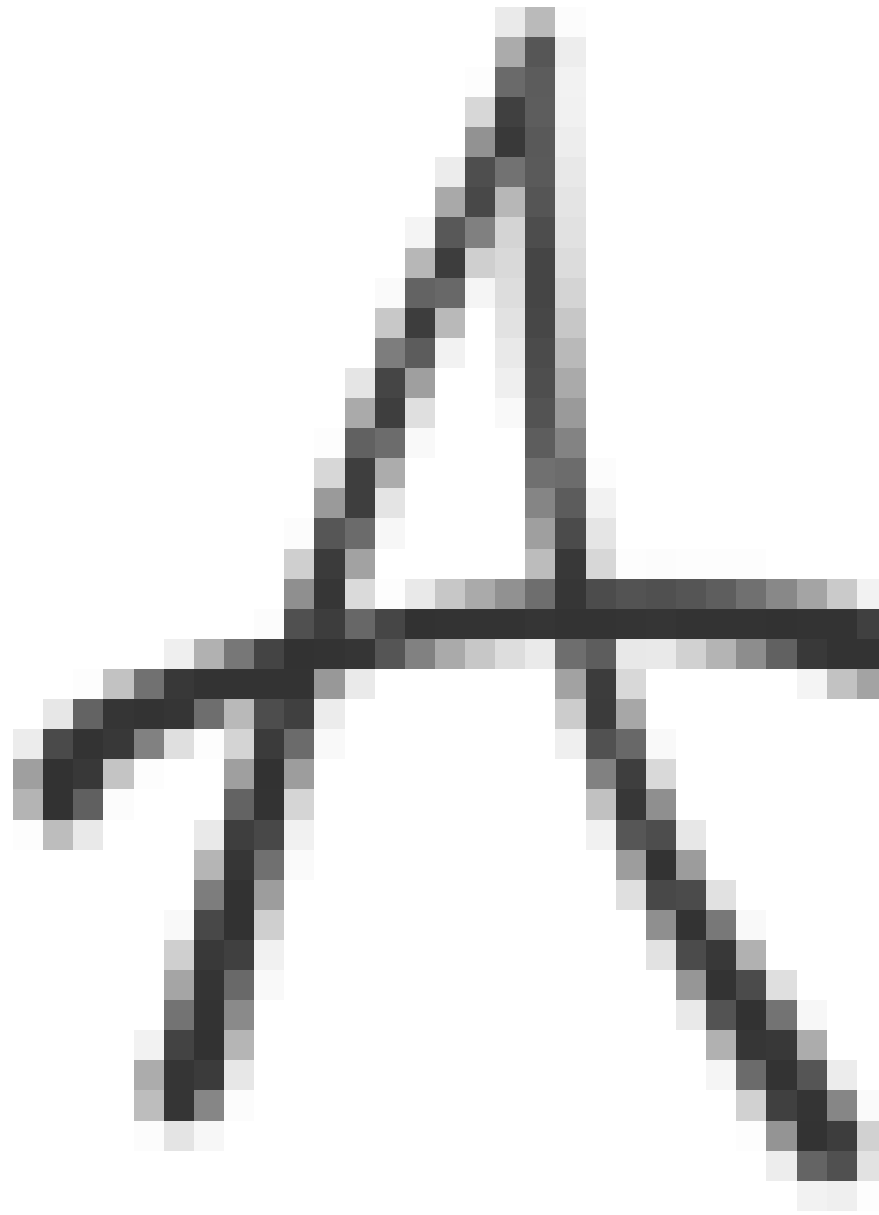






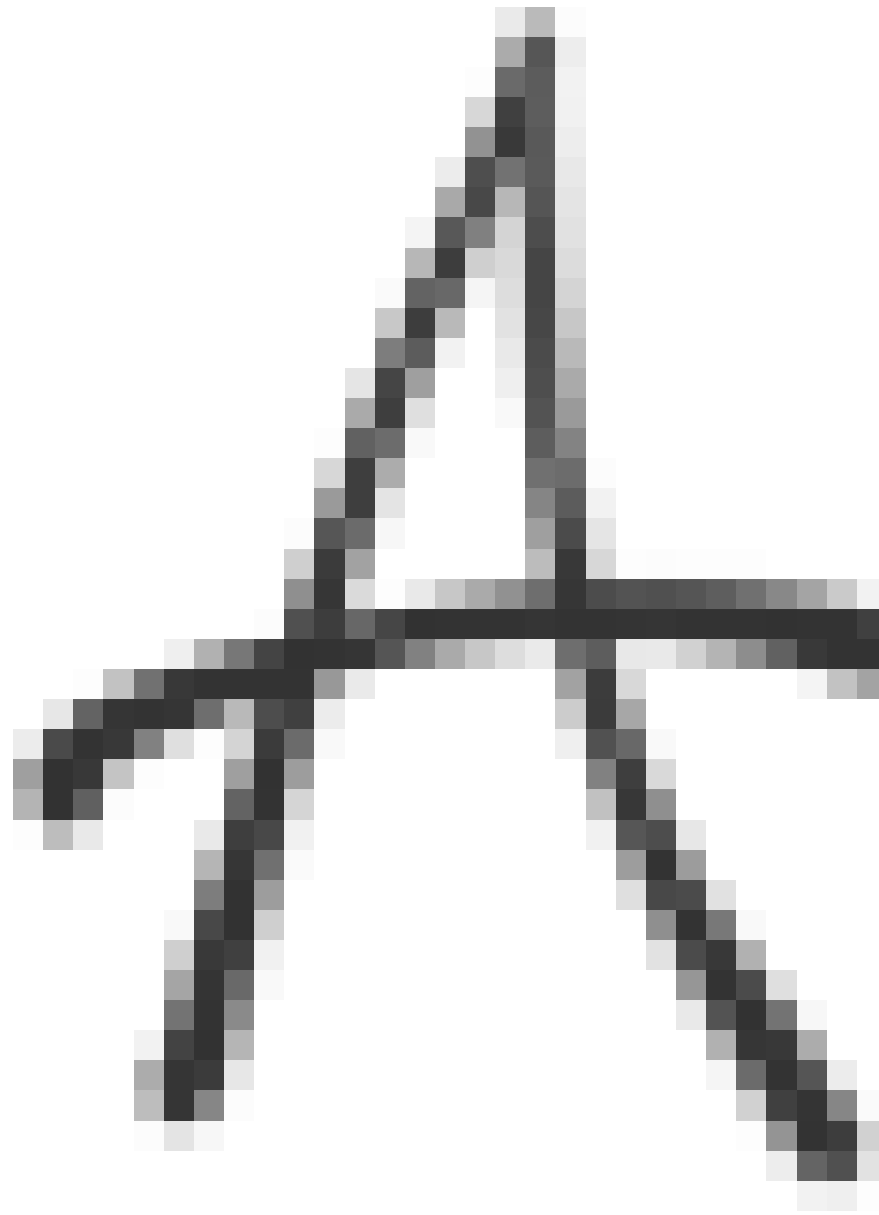


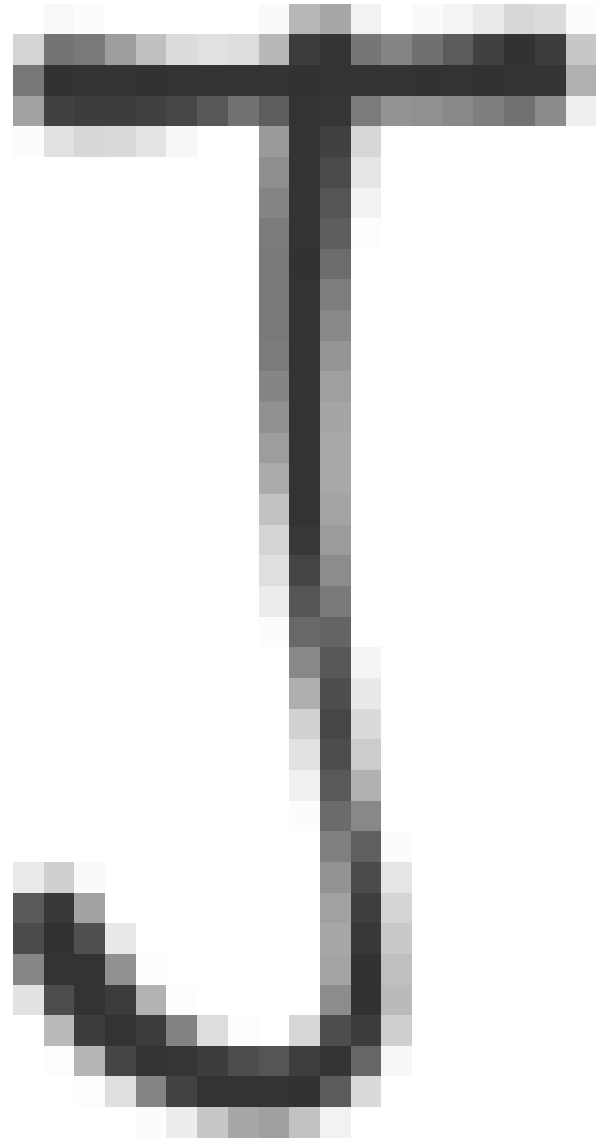


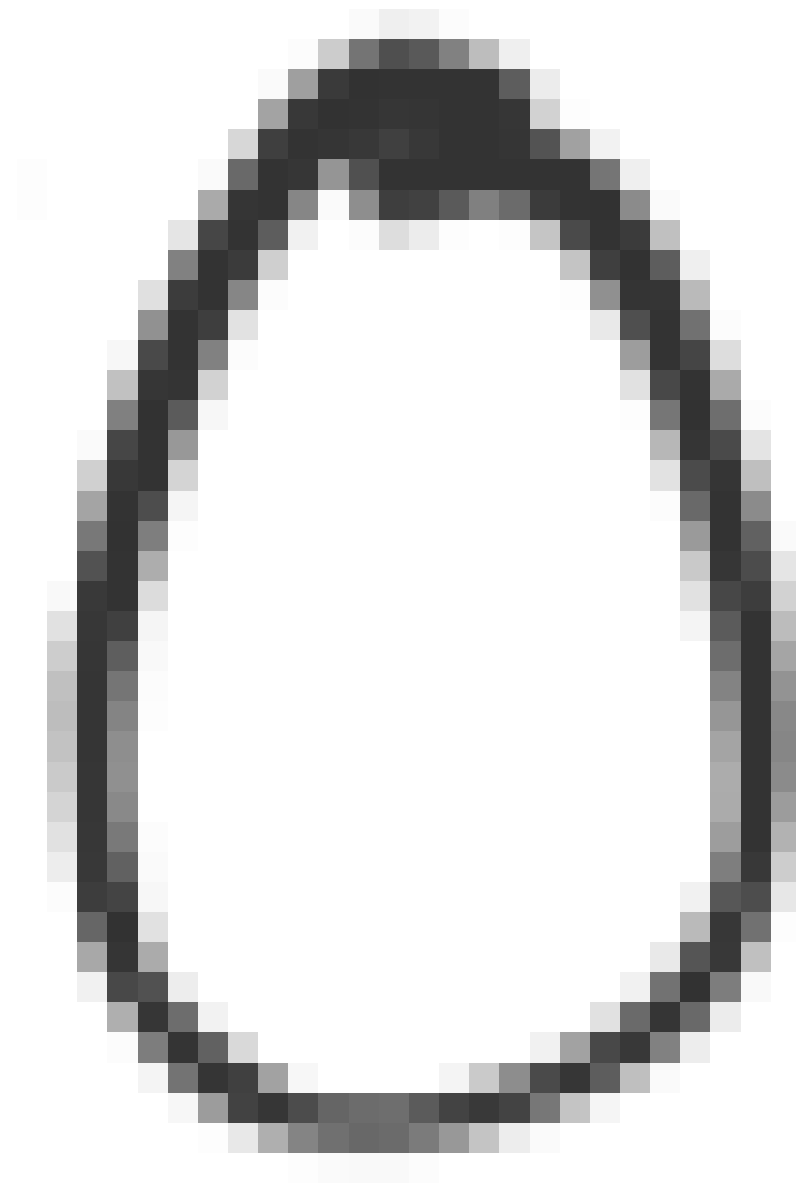


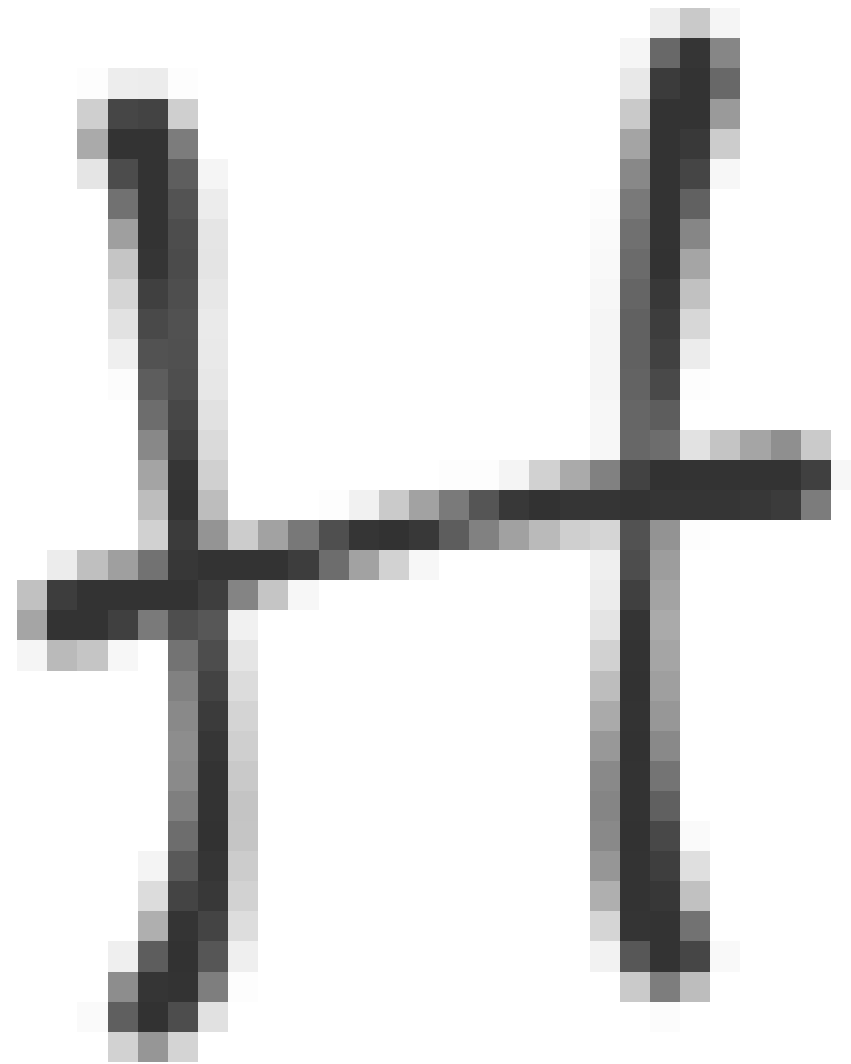
W



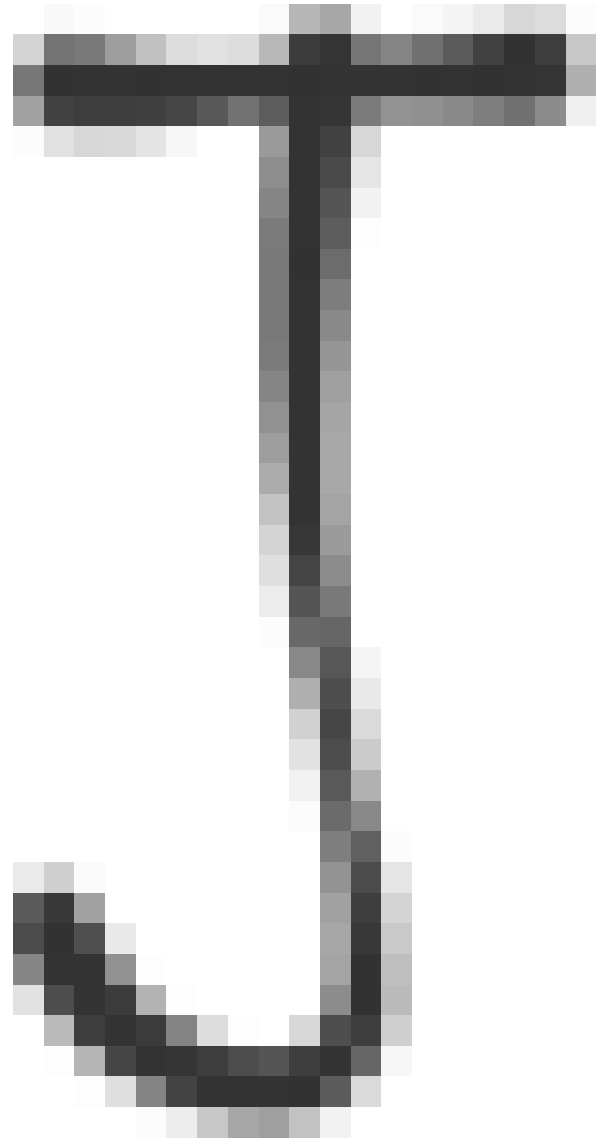


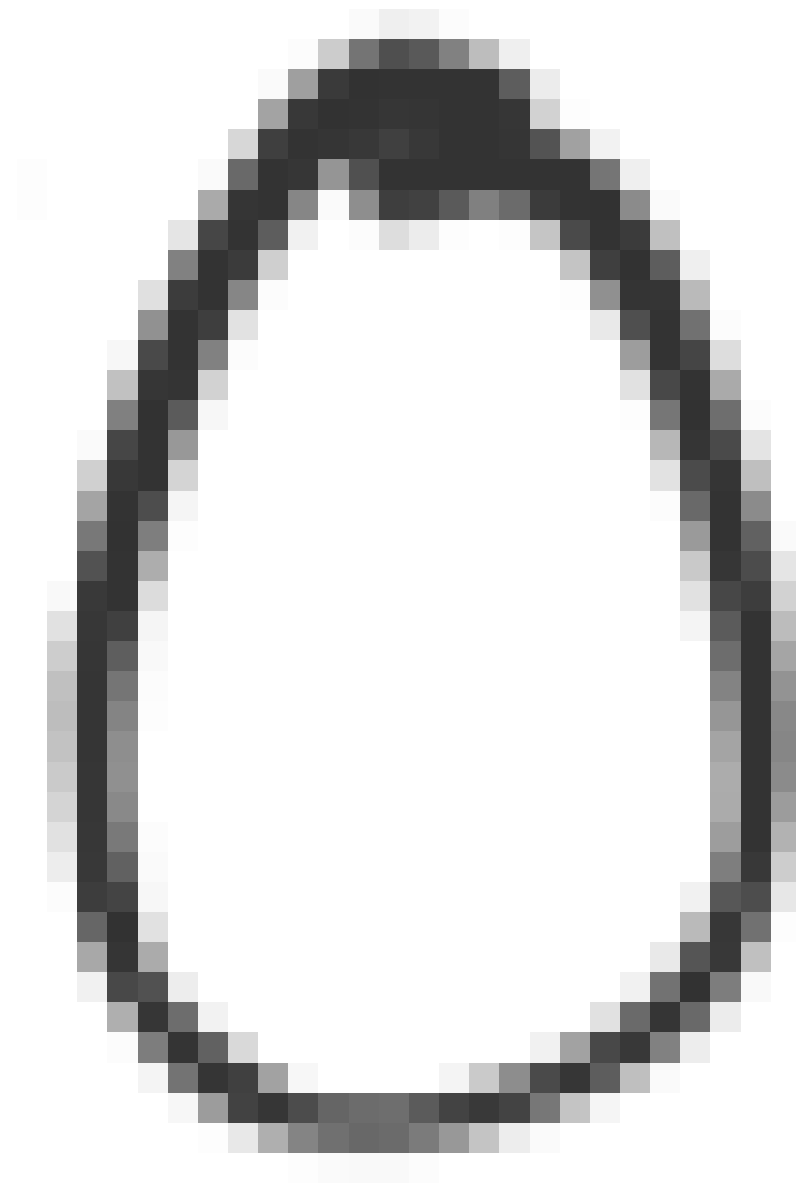


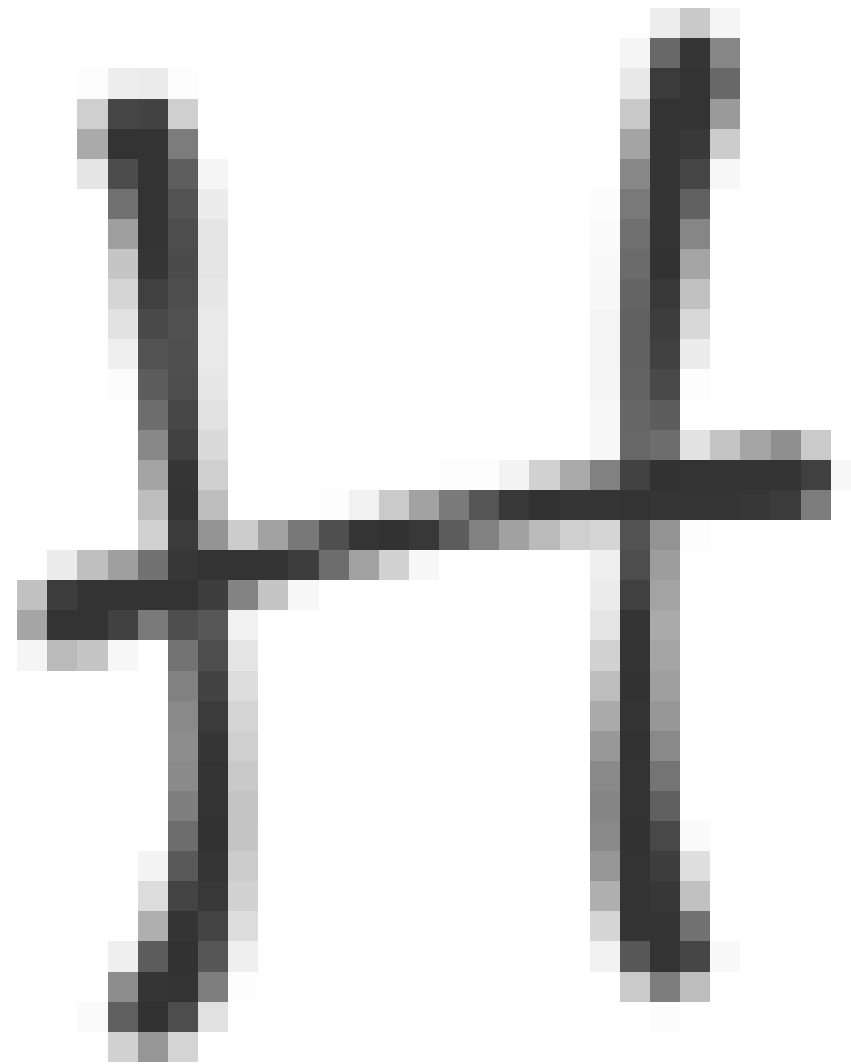




W

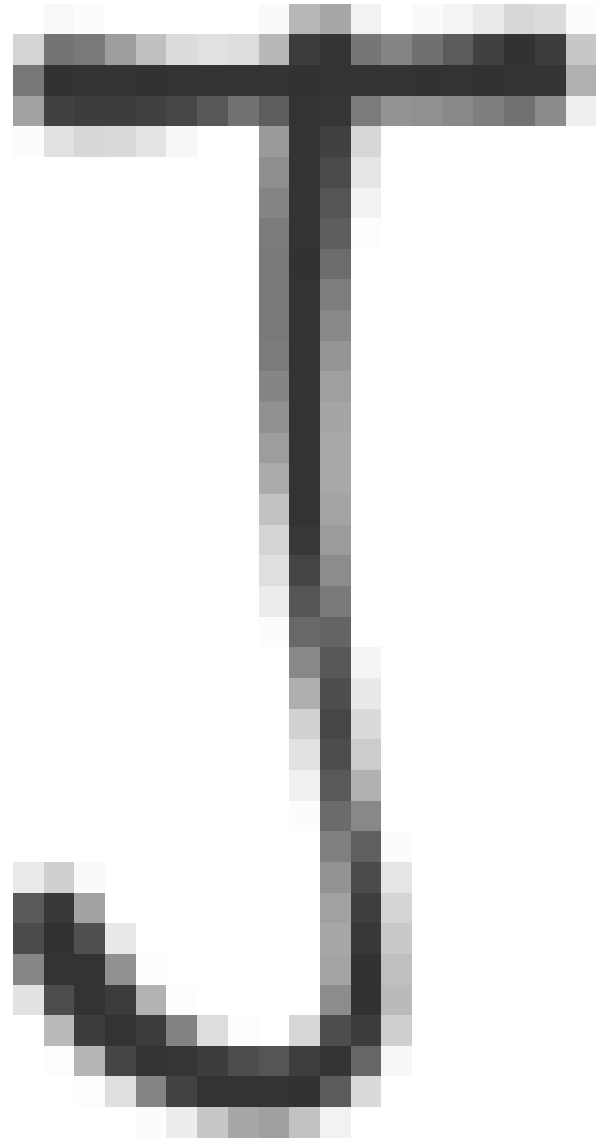


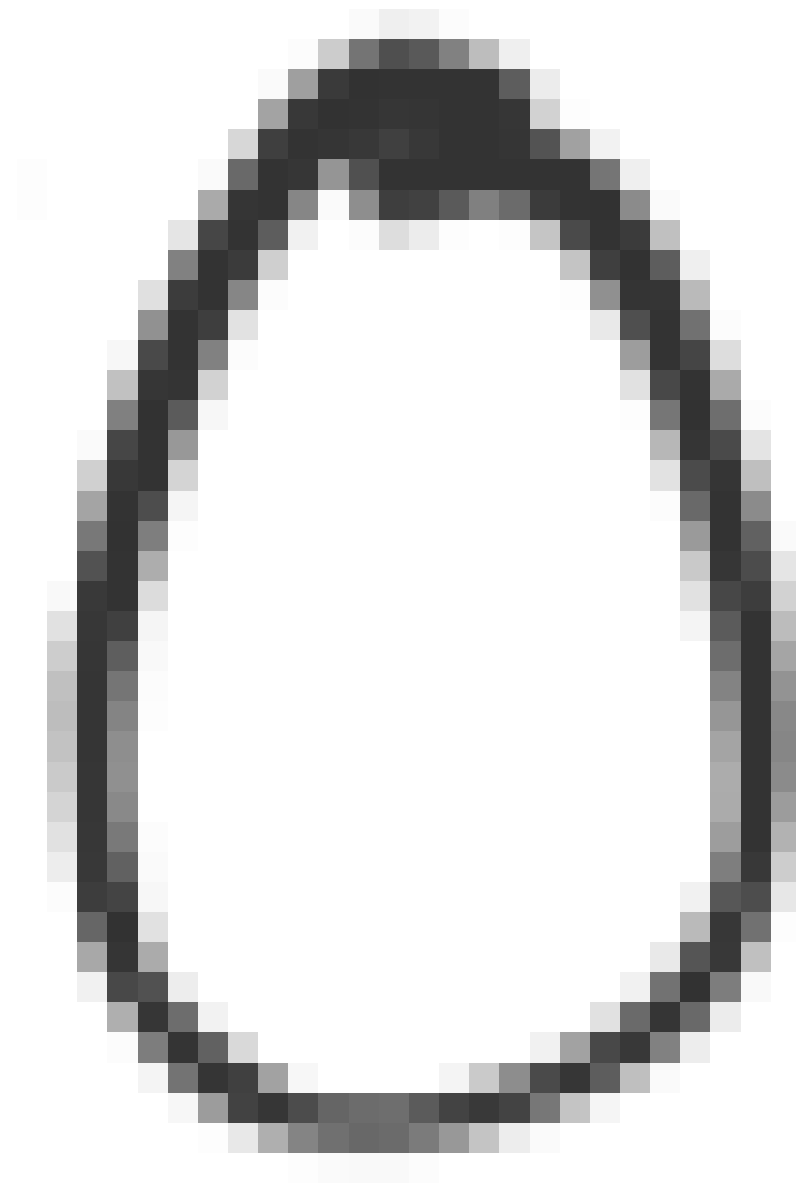


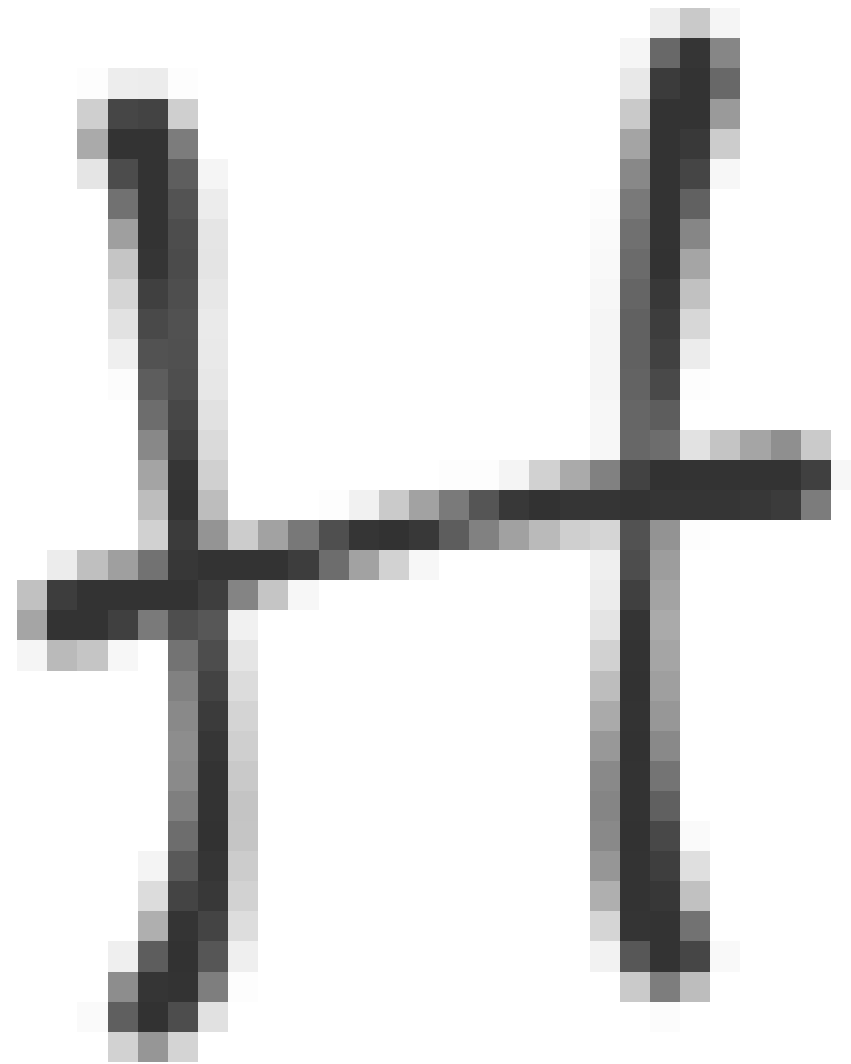




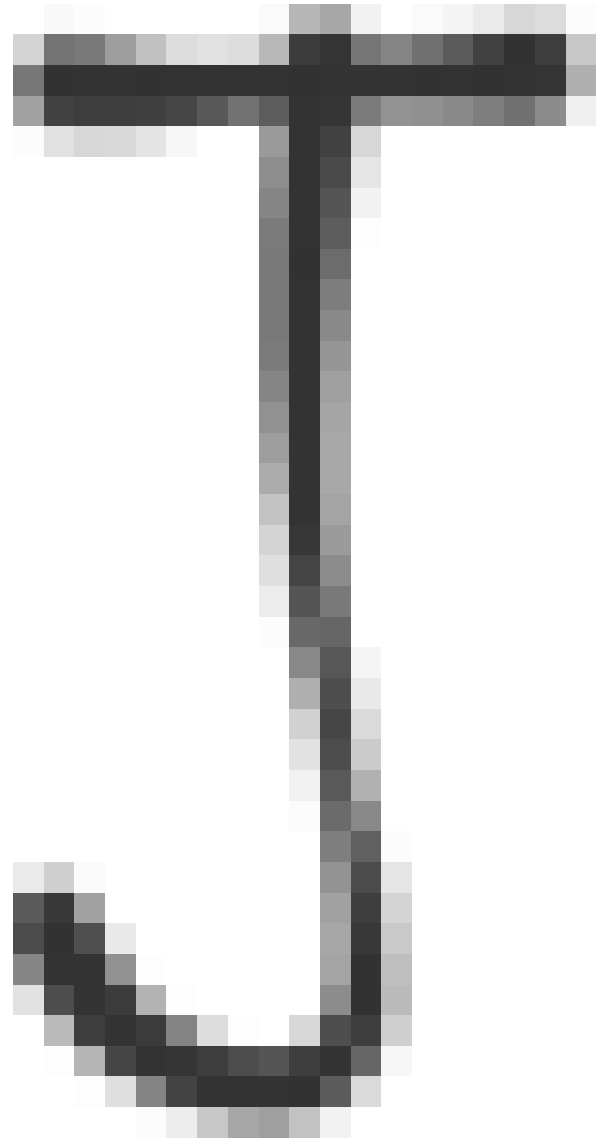
W

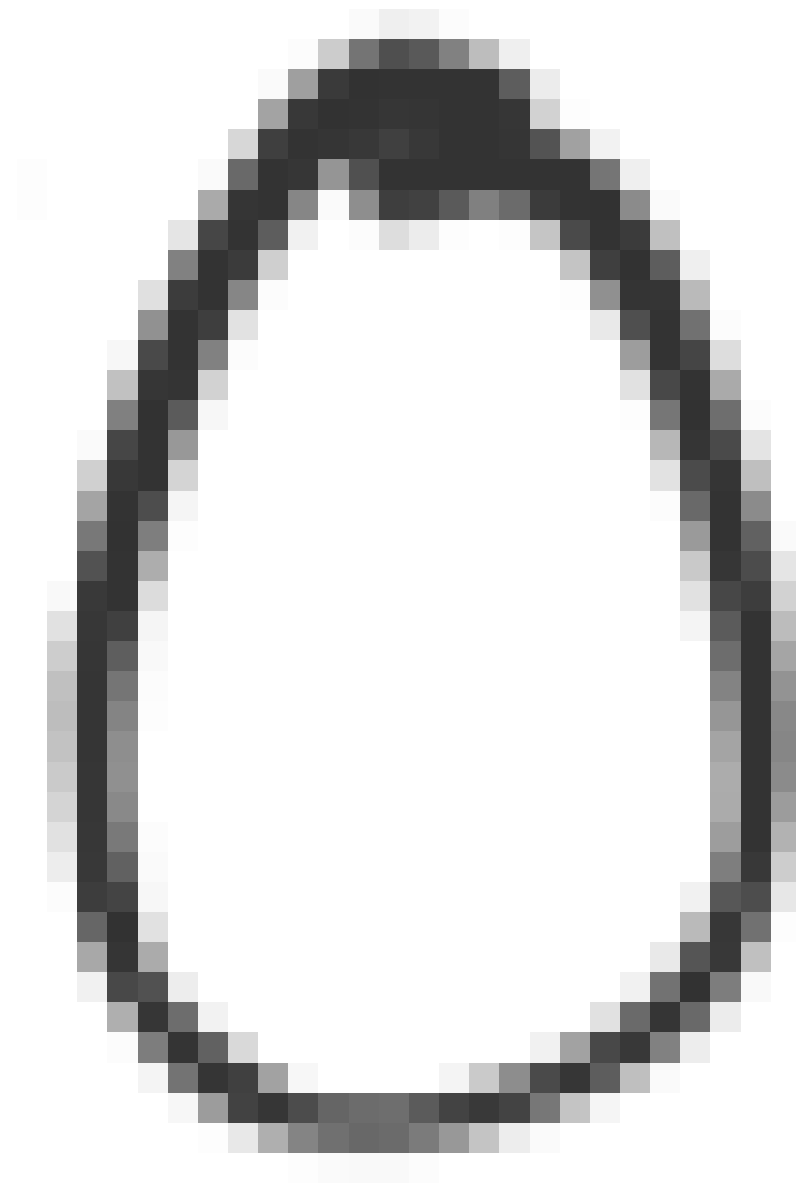


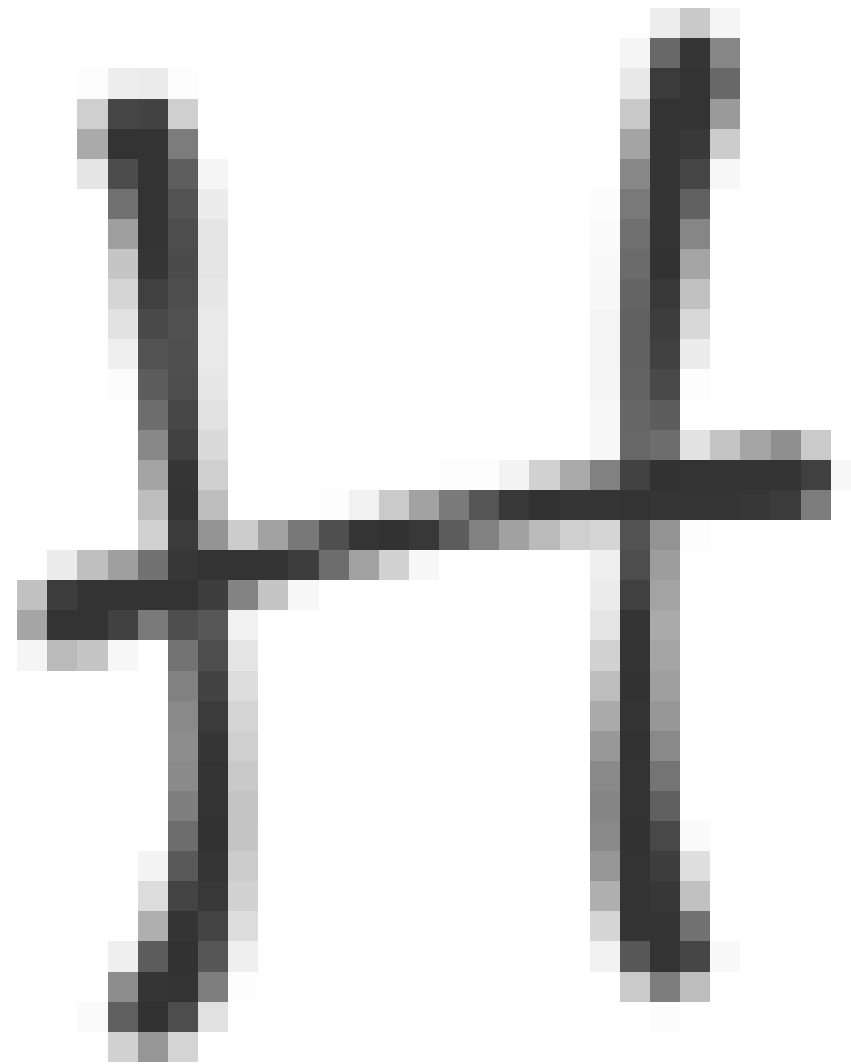




W

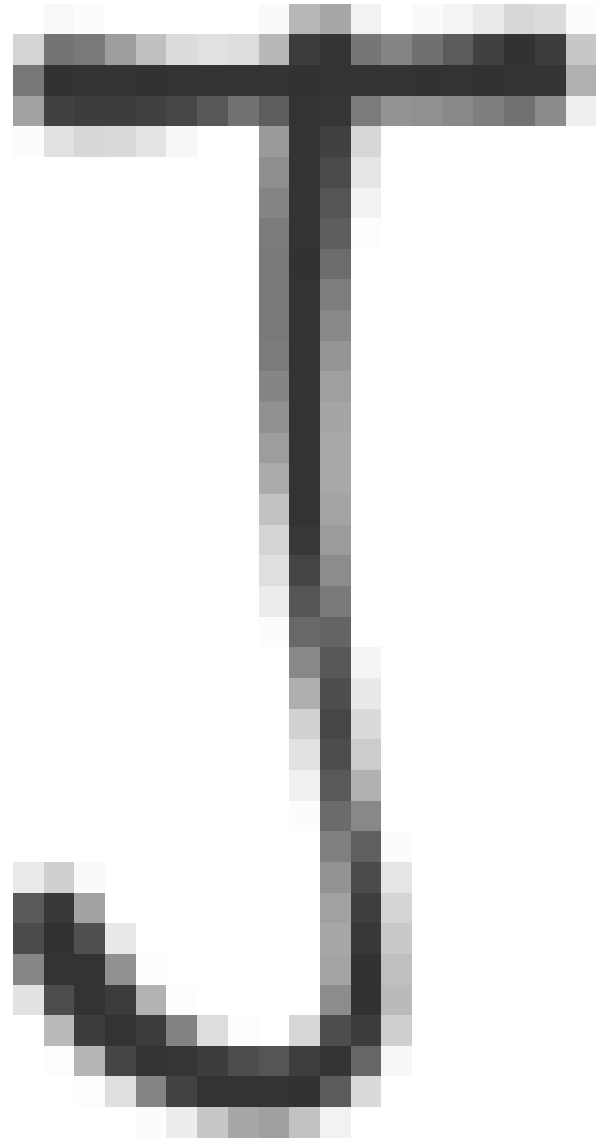


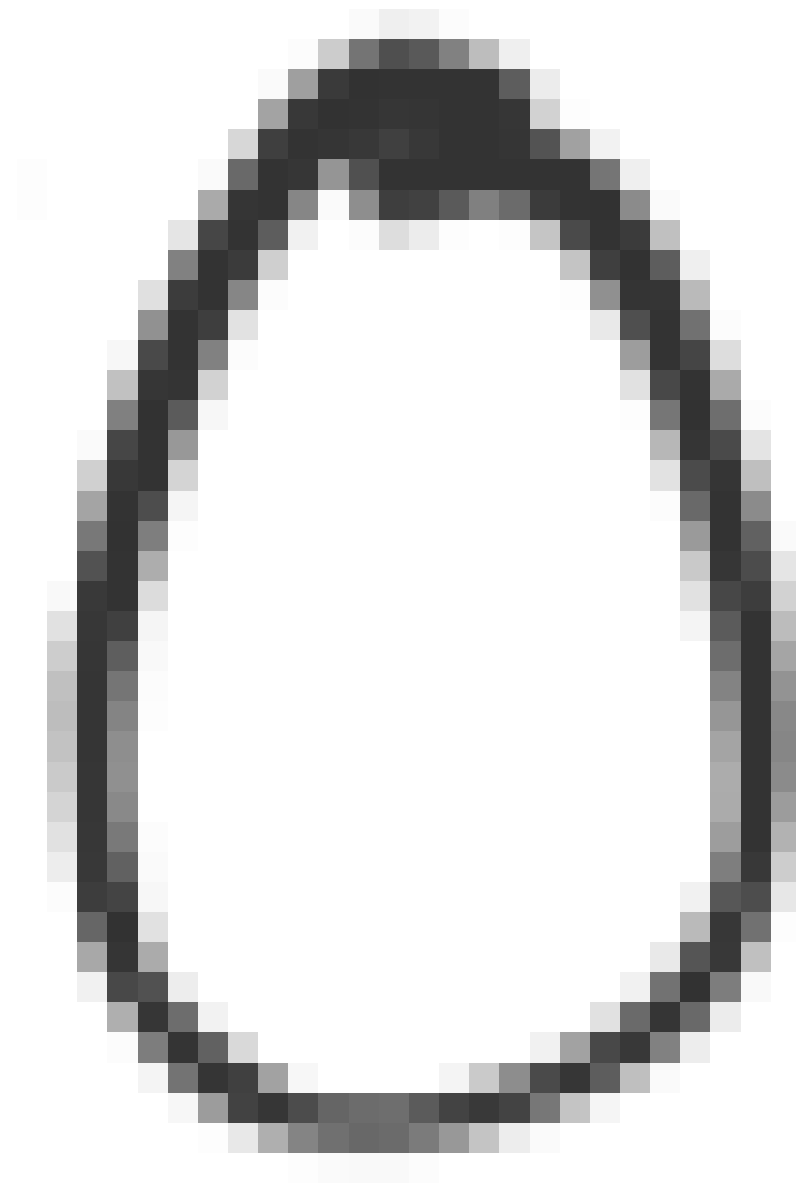


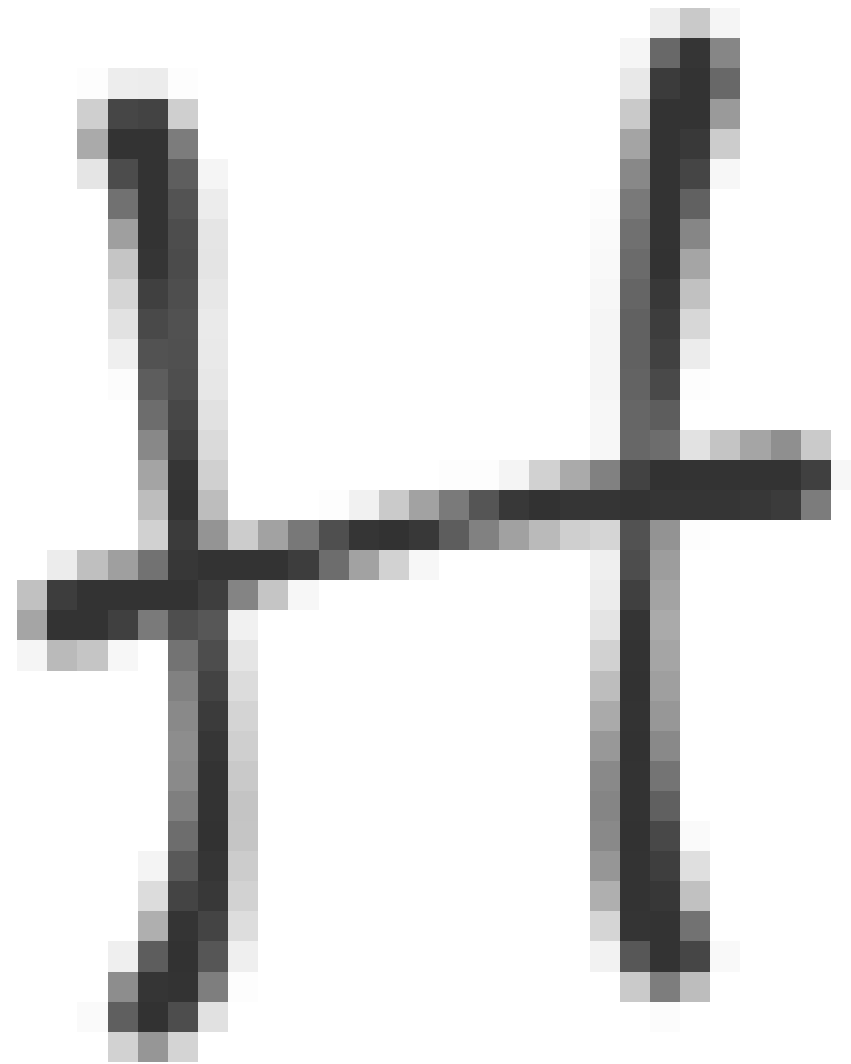




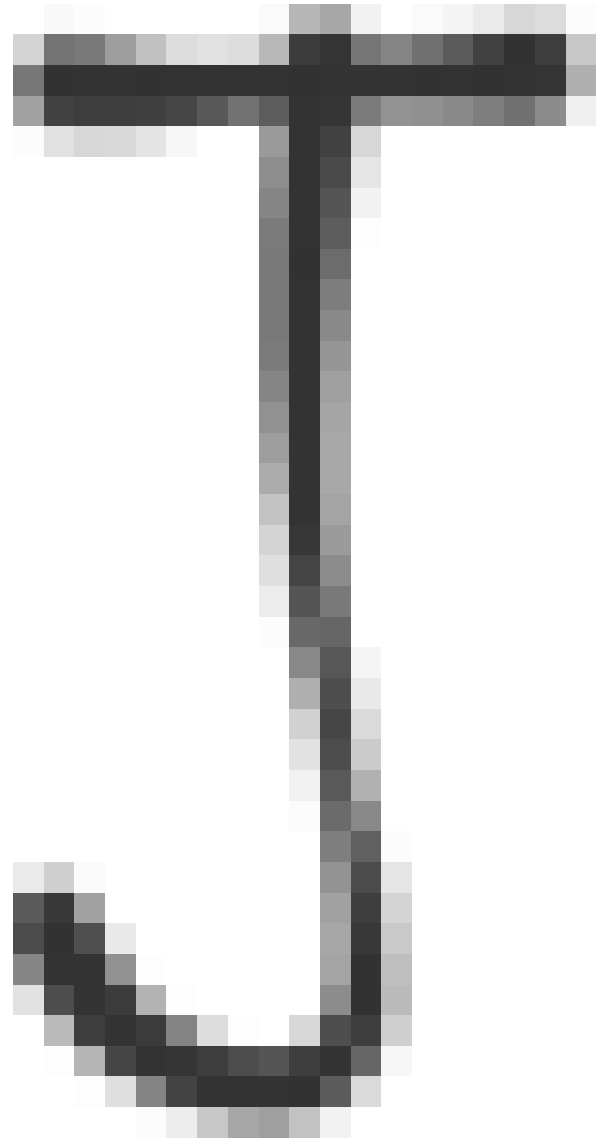
W

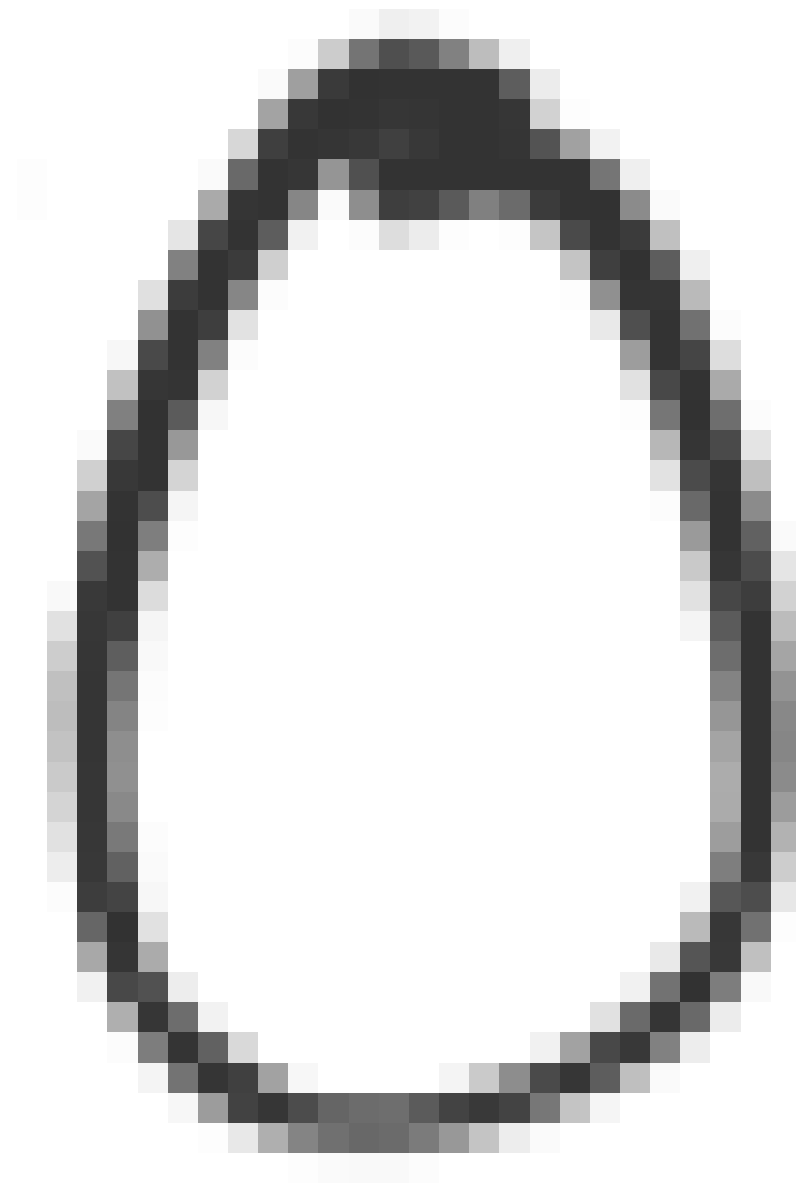


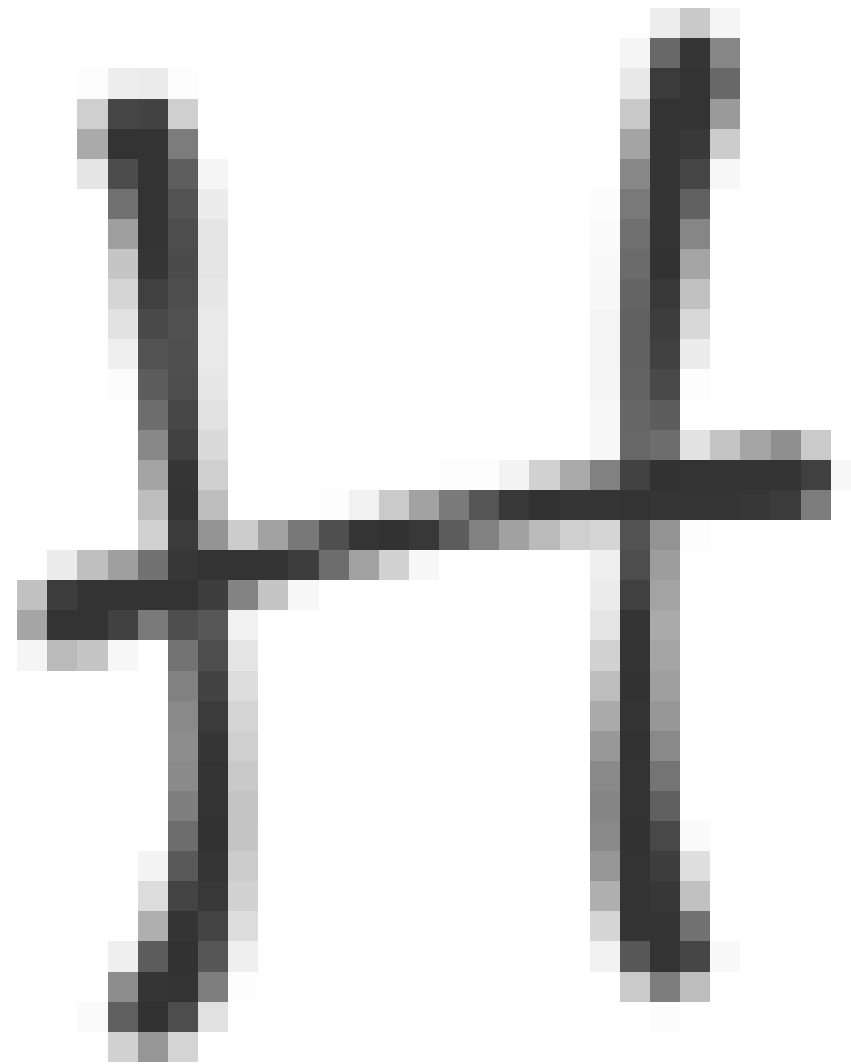




W

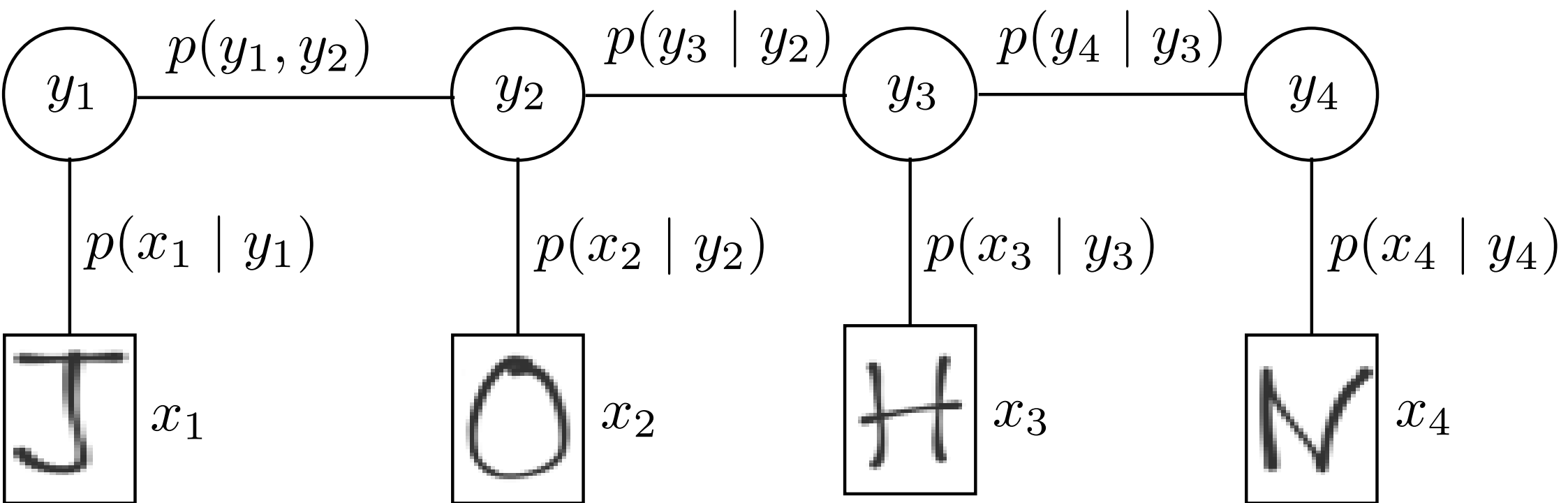


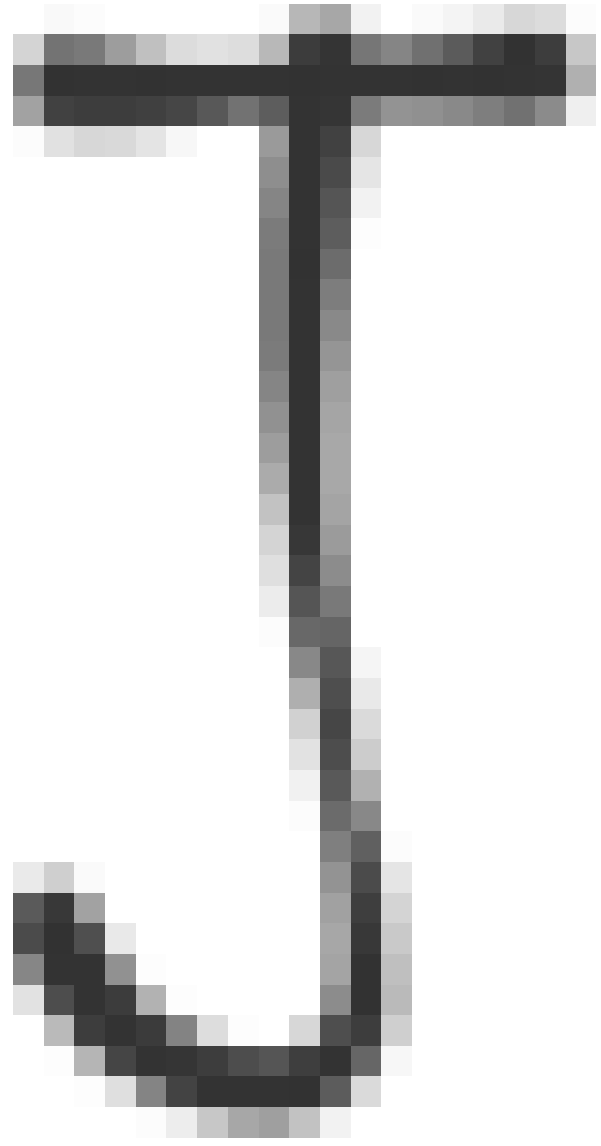


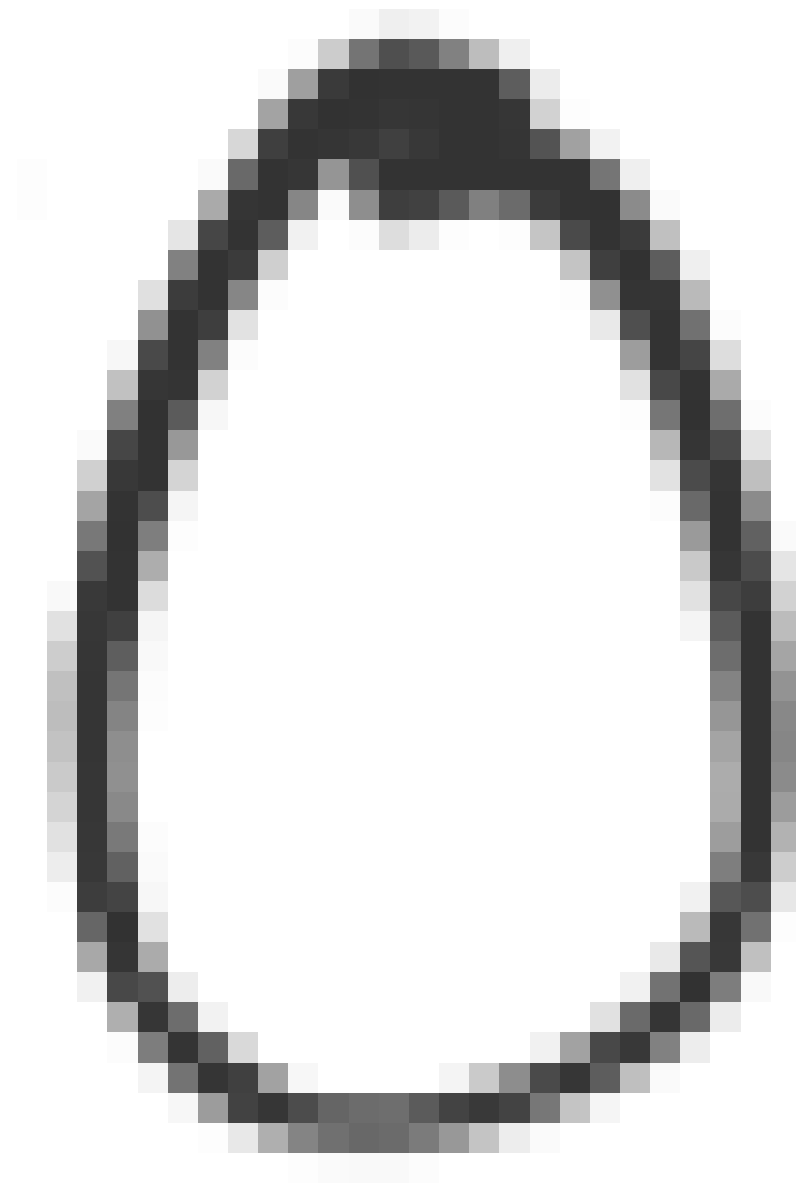


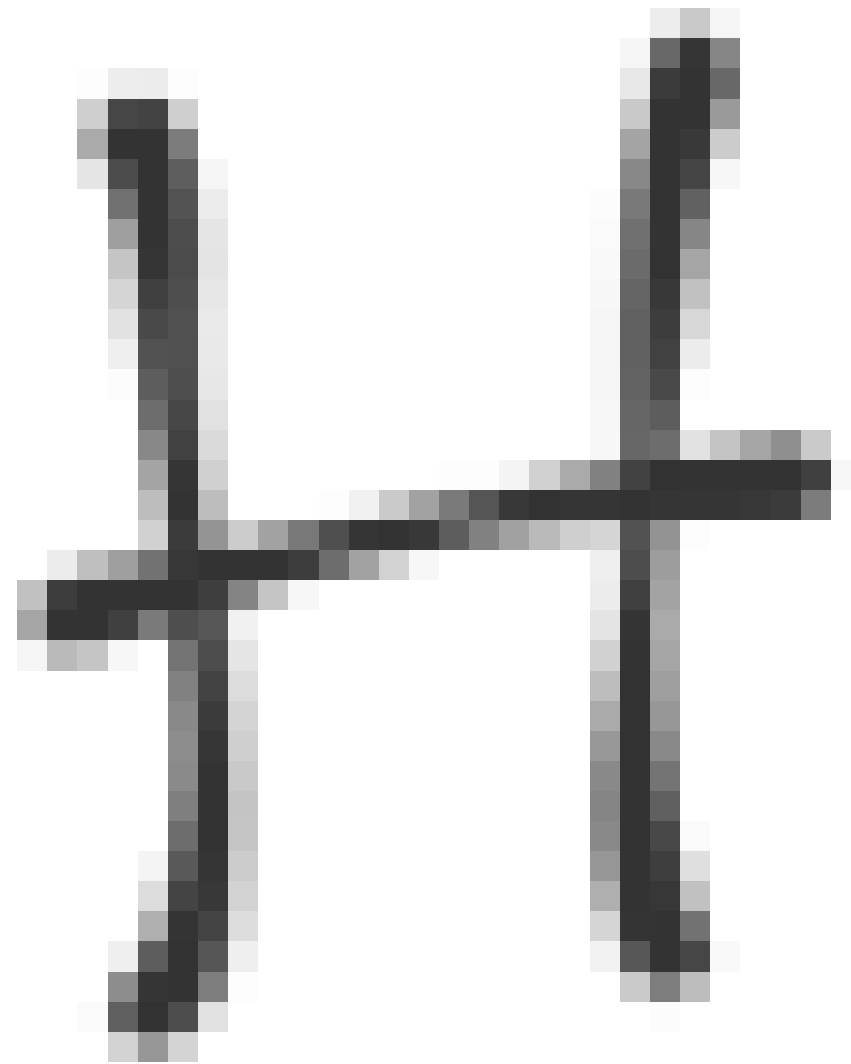


W



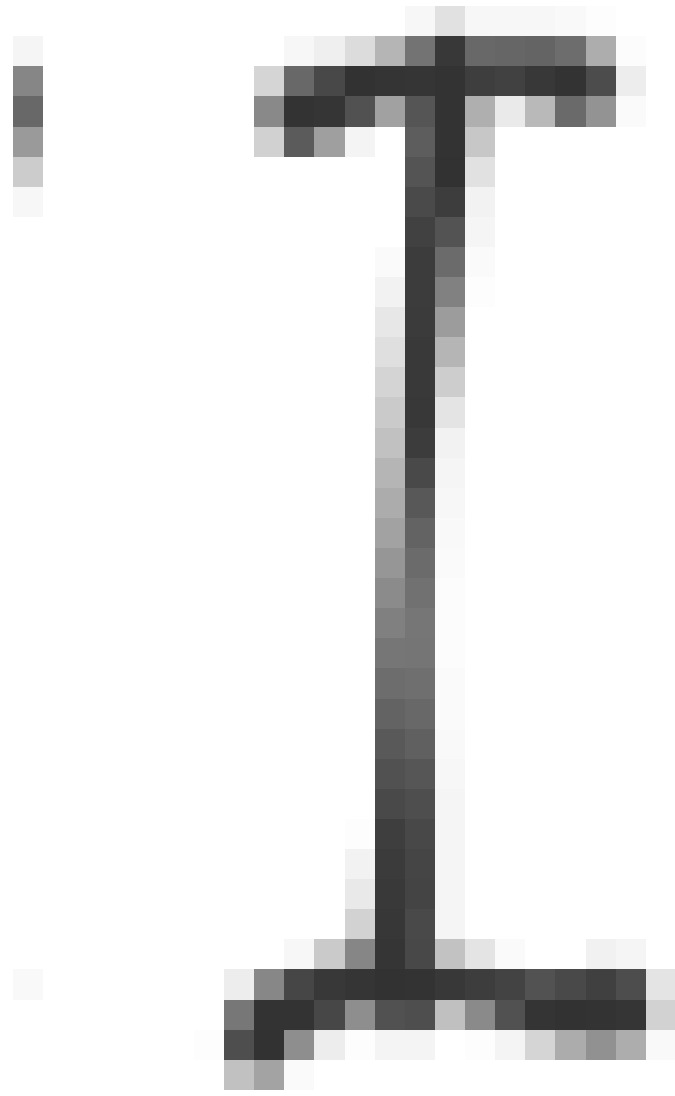




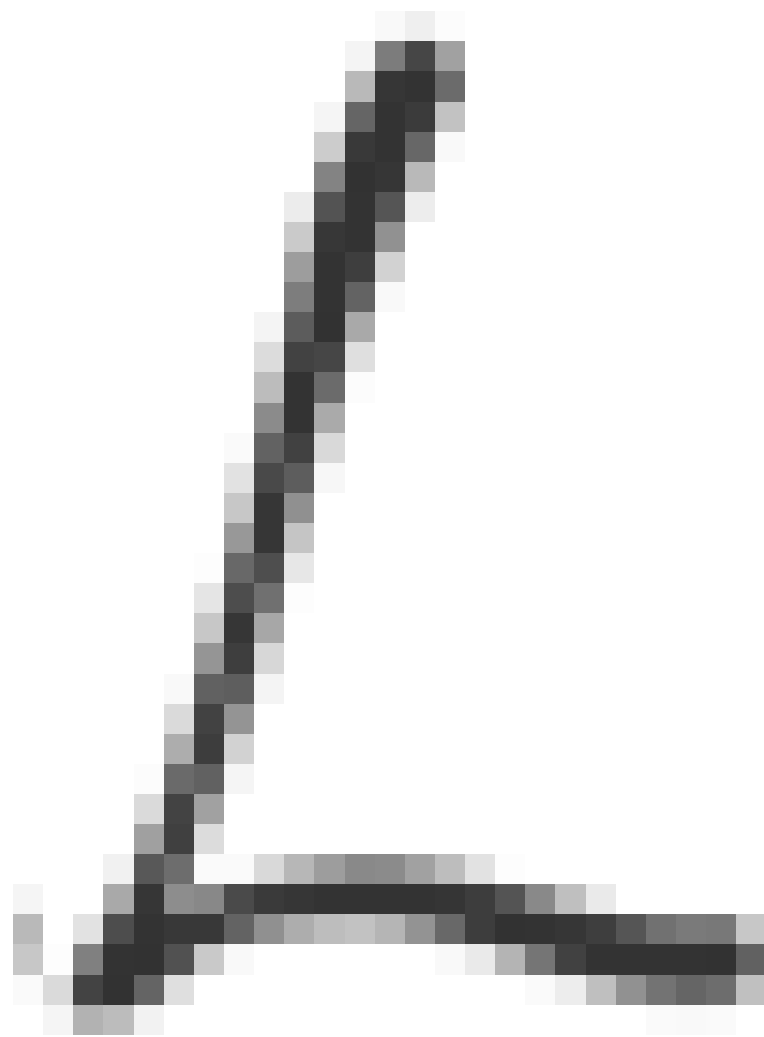


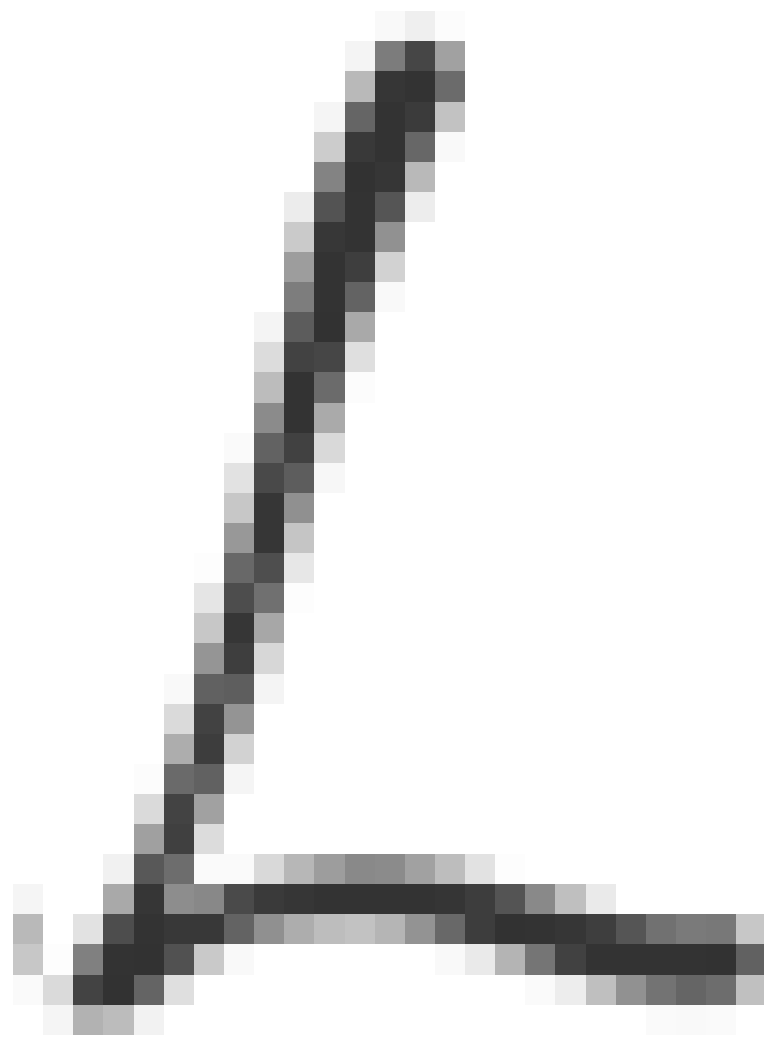
W

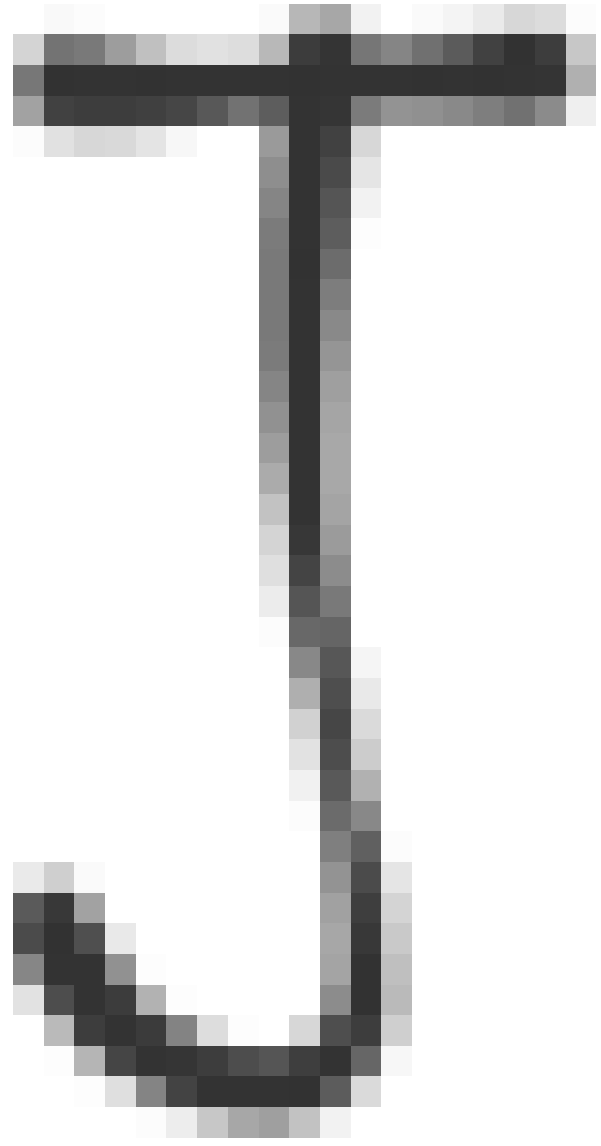


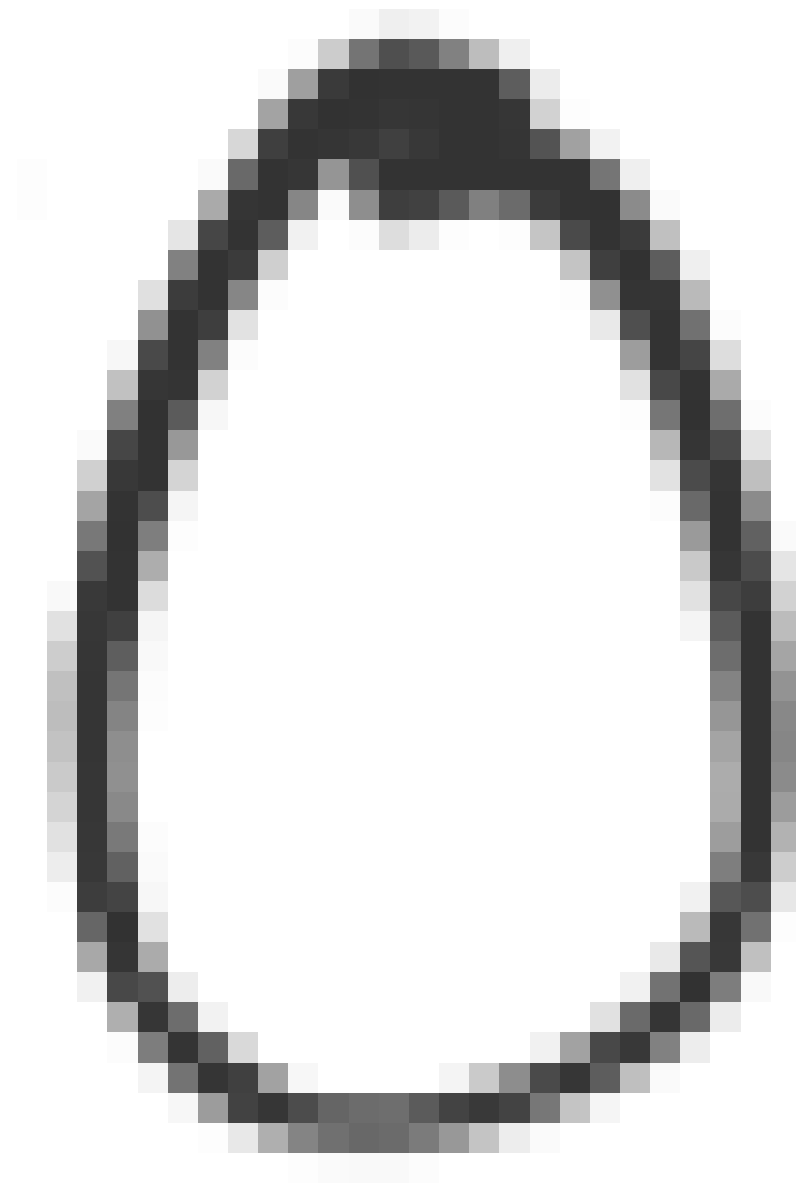


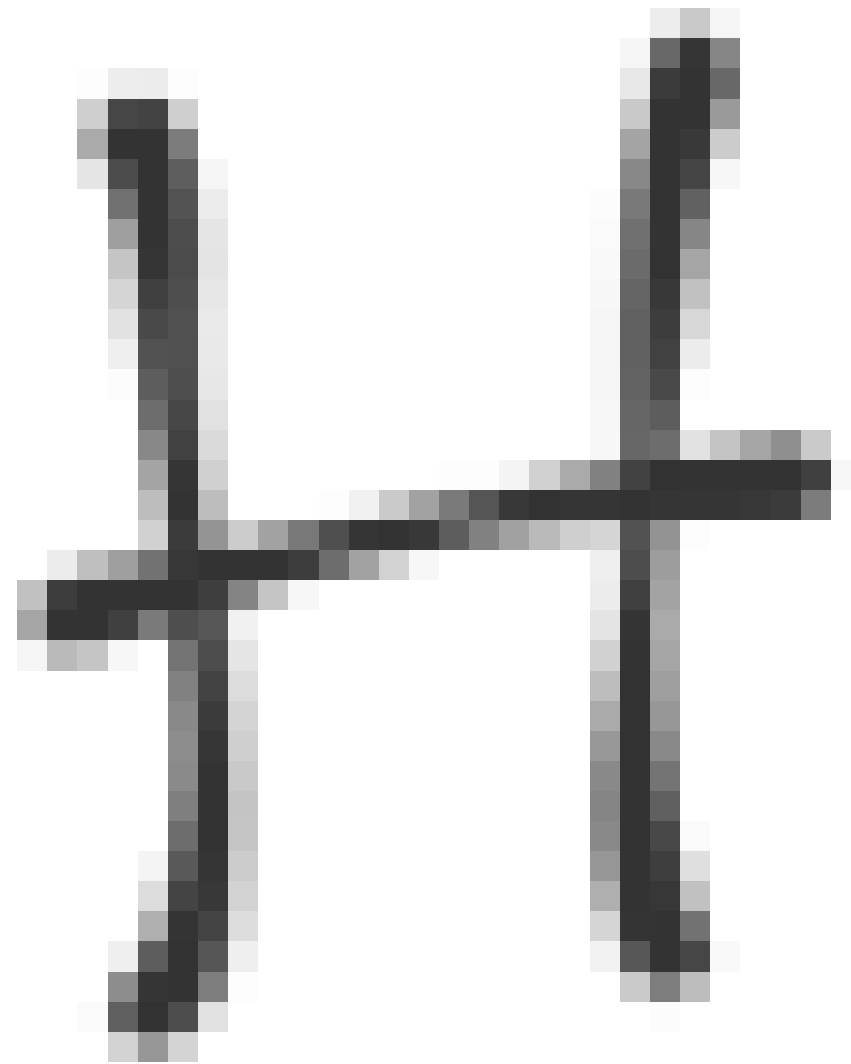




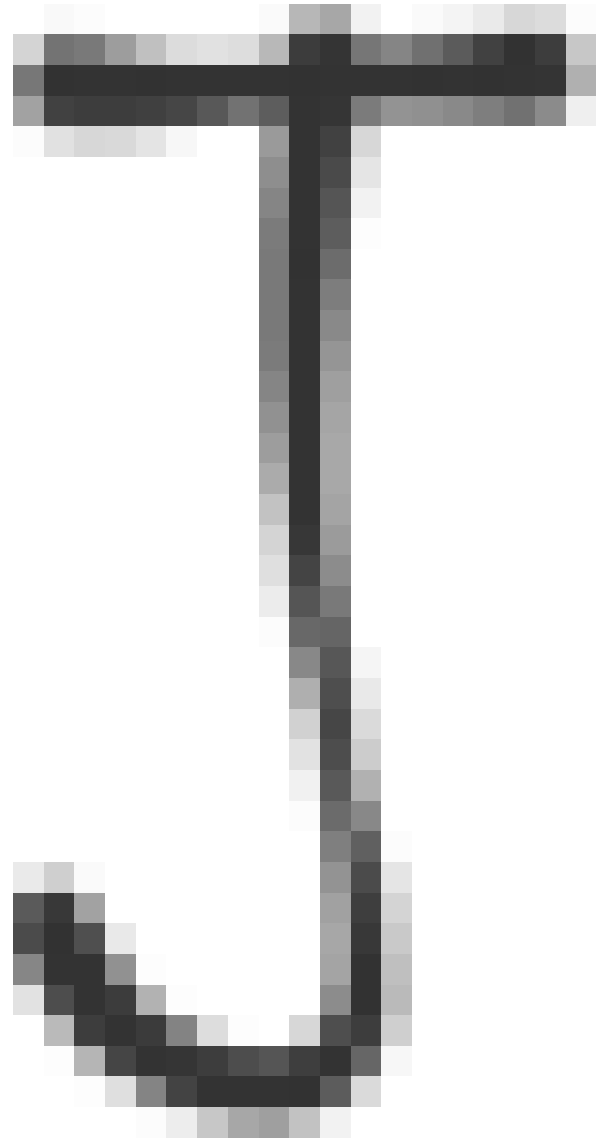


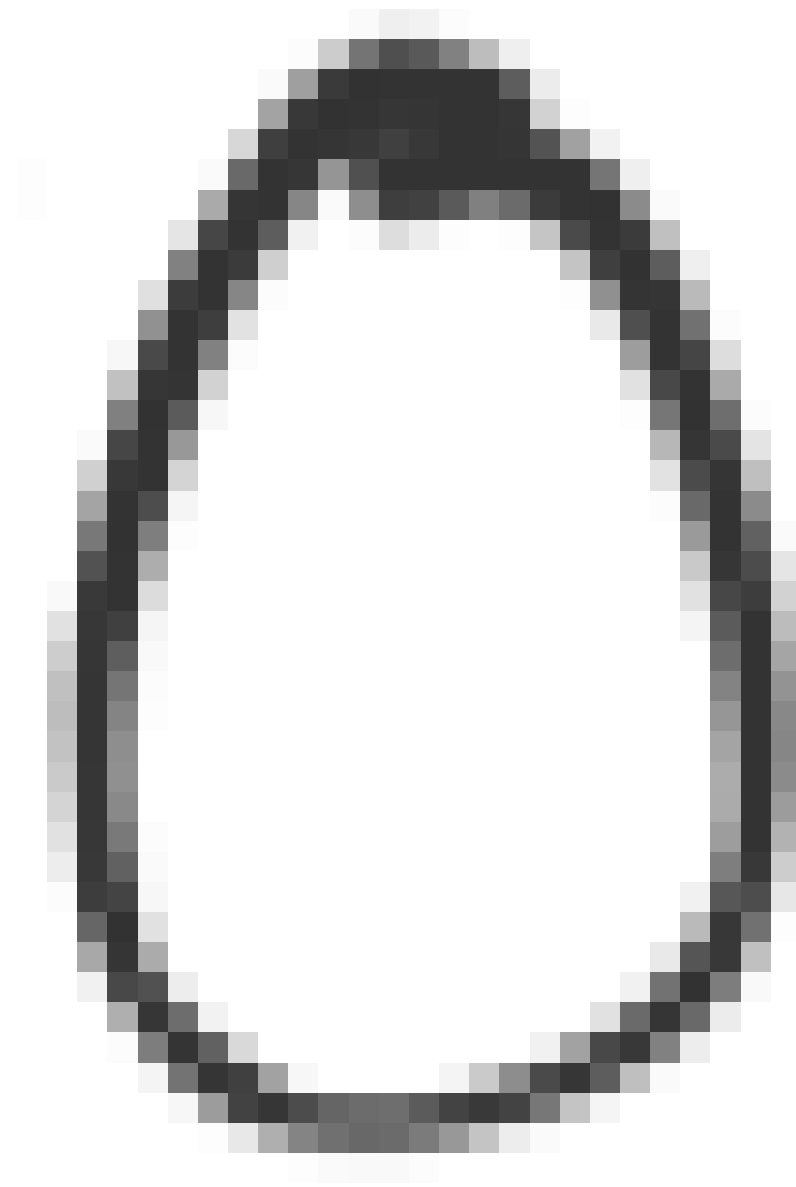




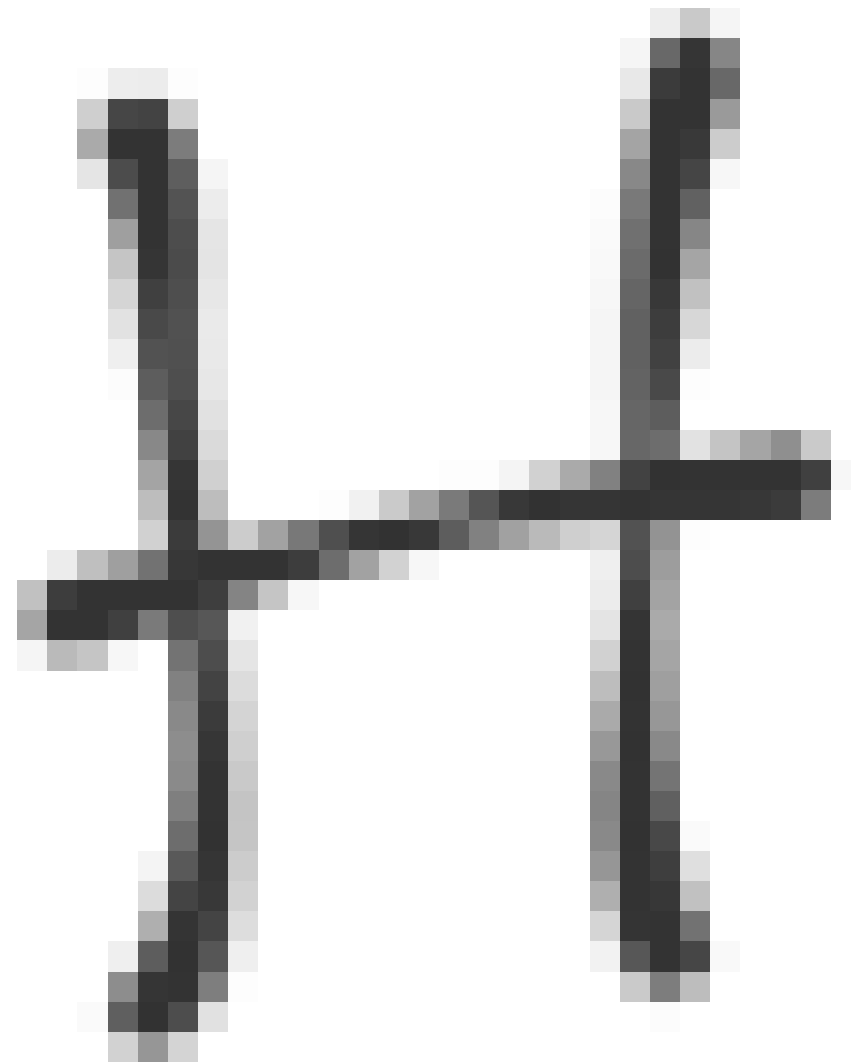


W

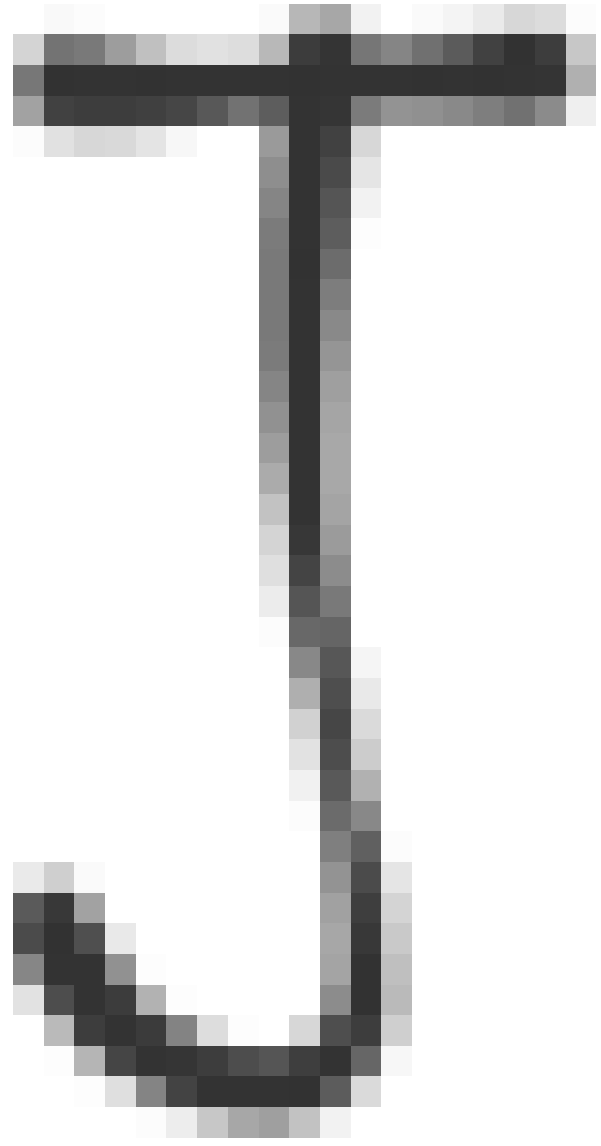


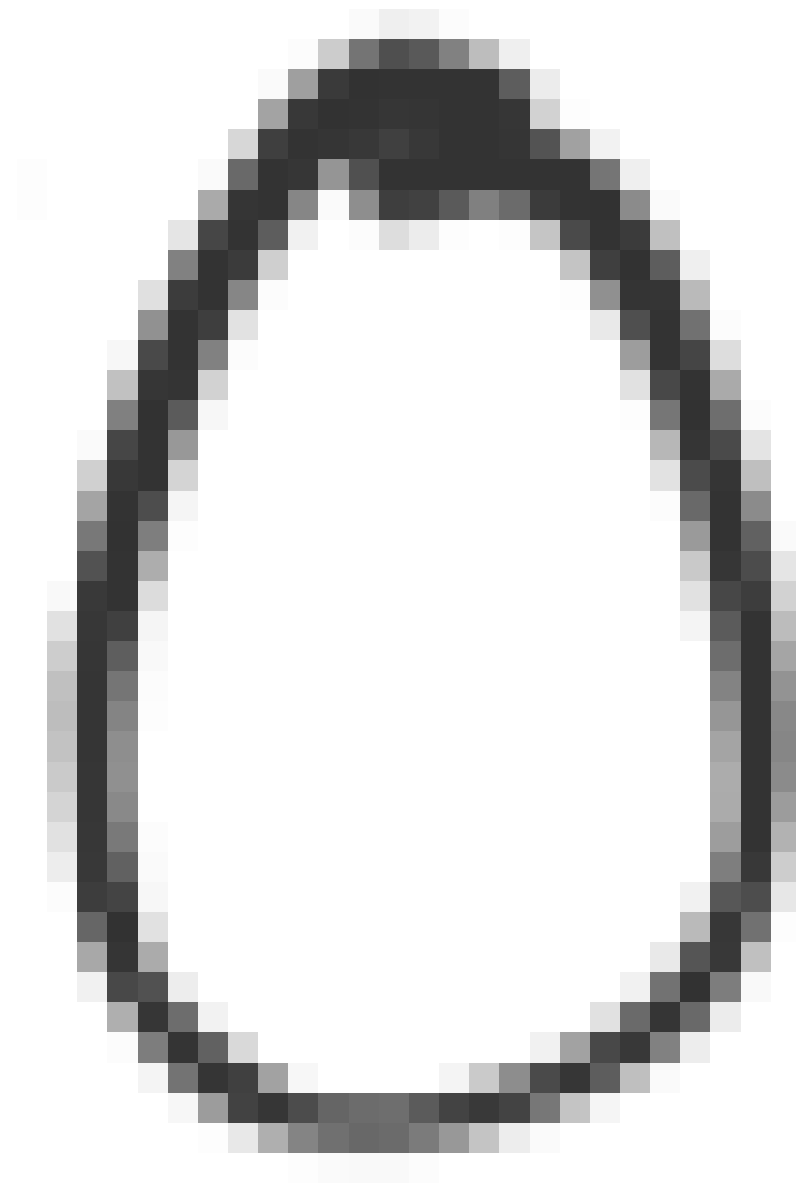


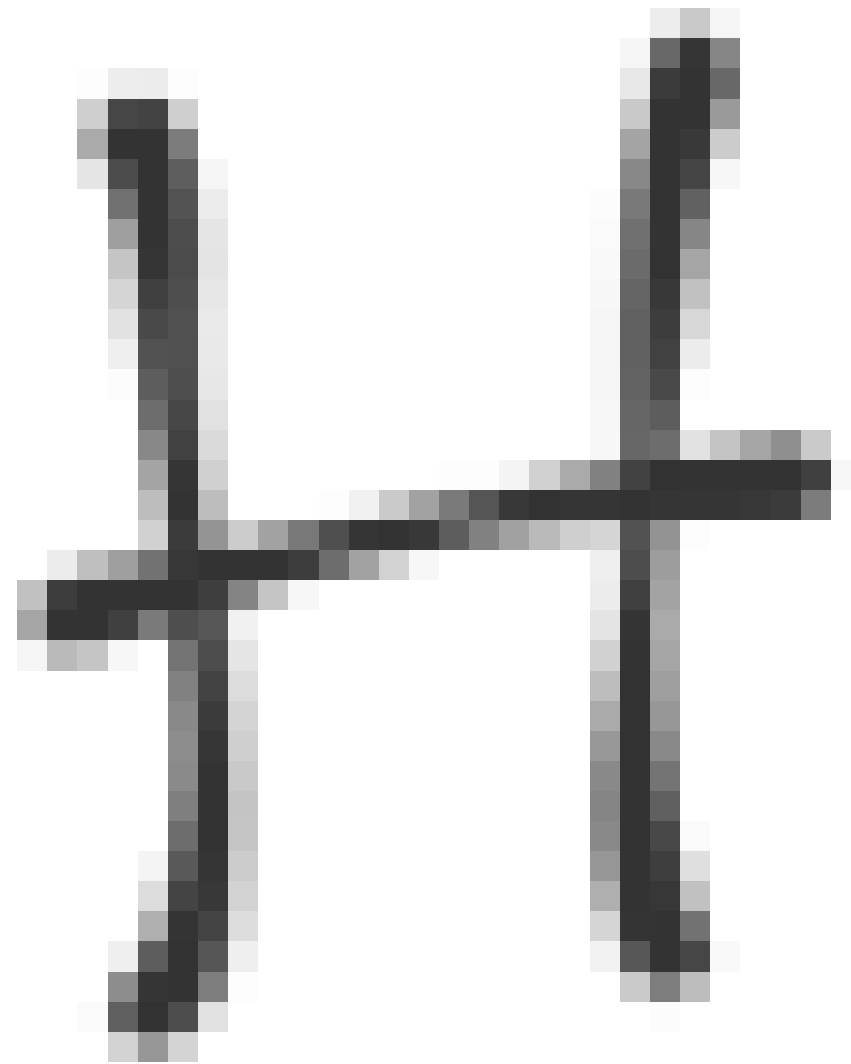




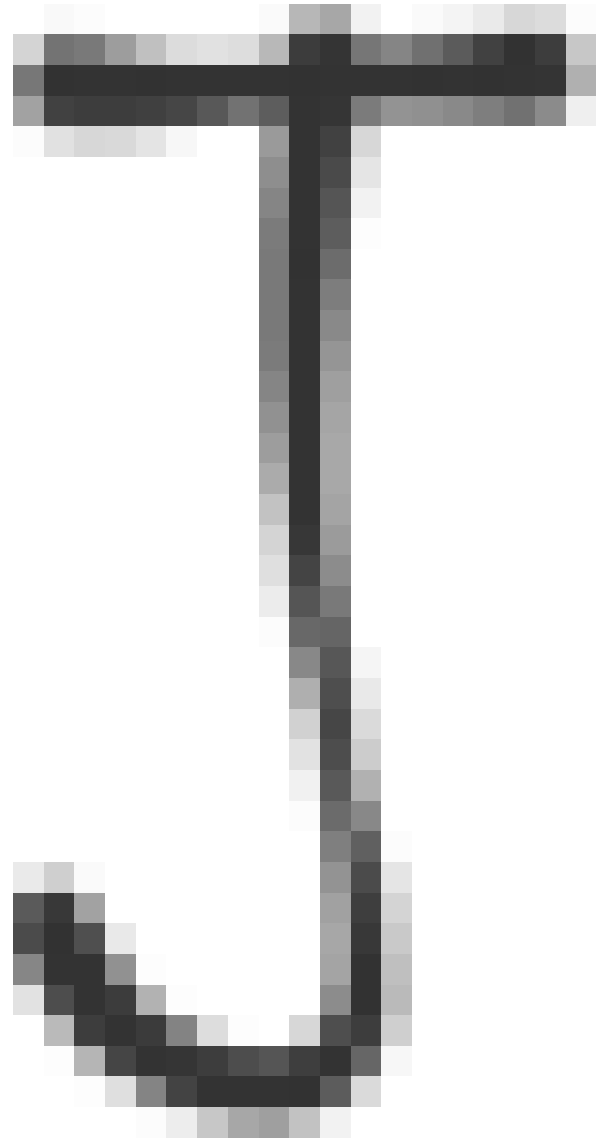
W

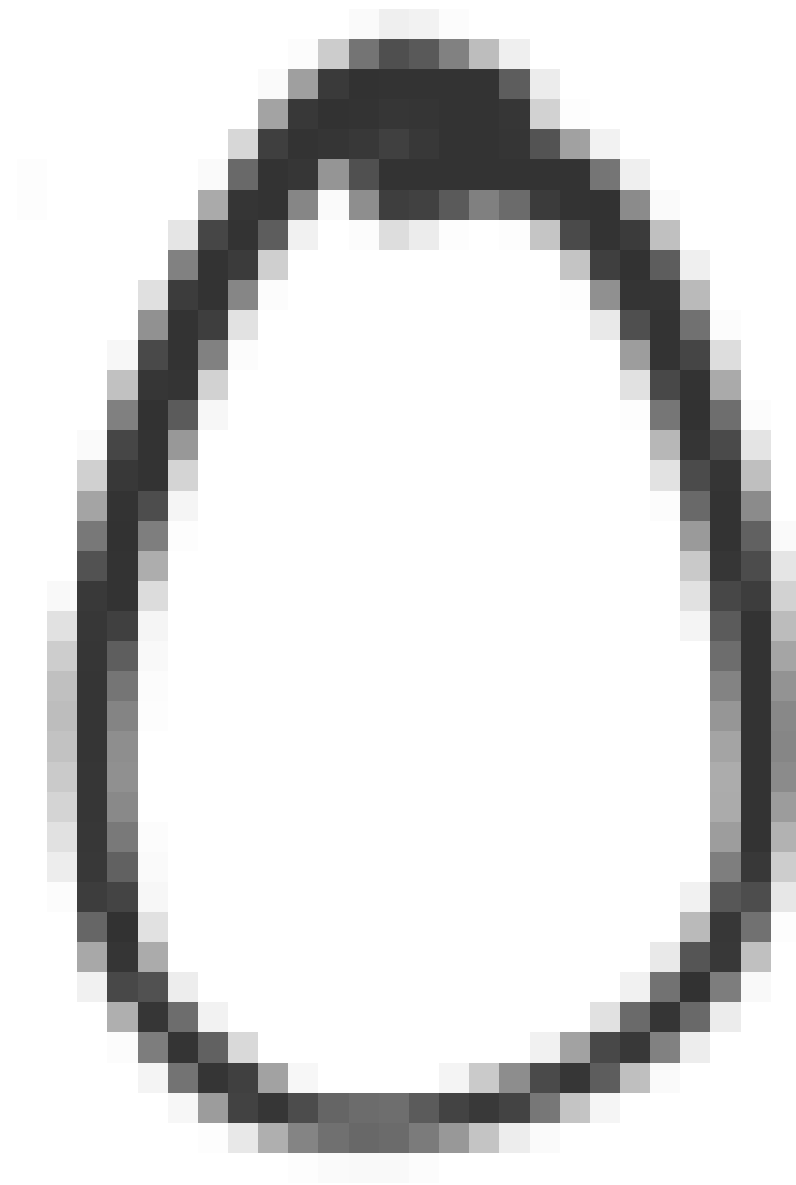




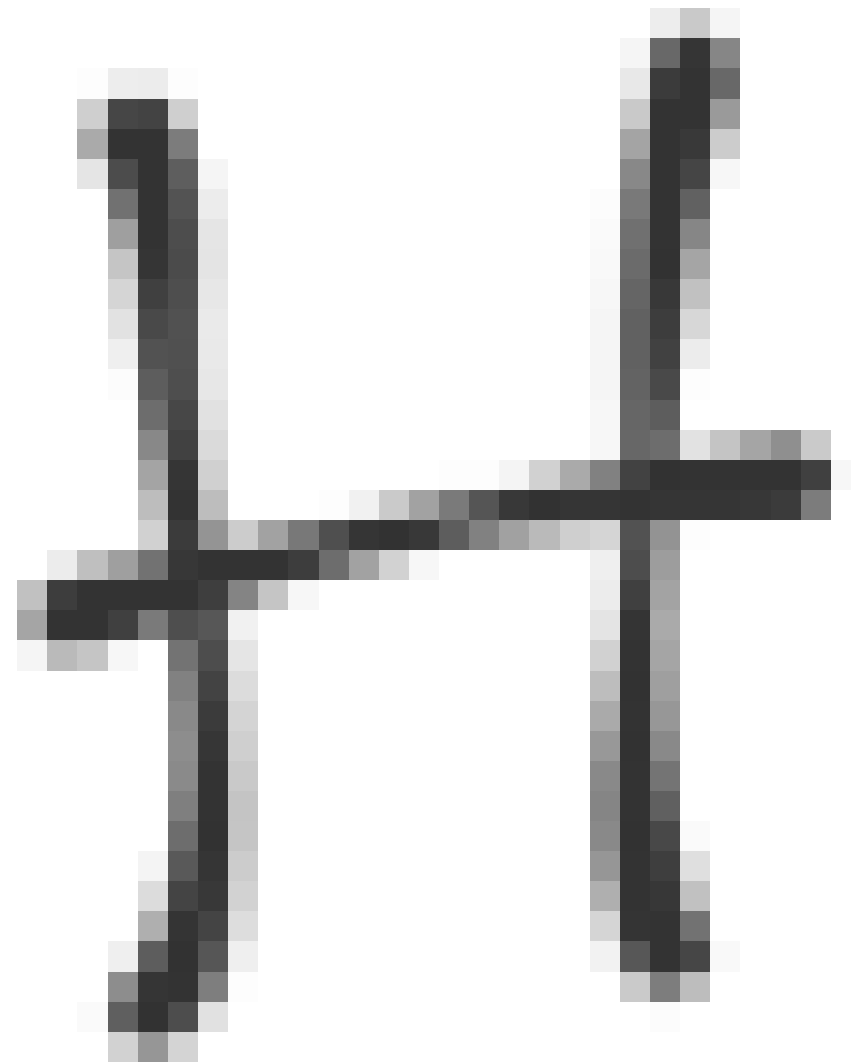


W

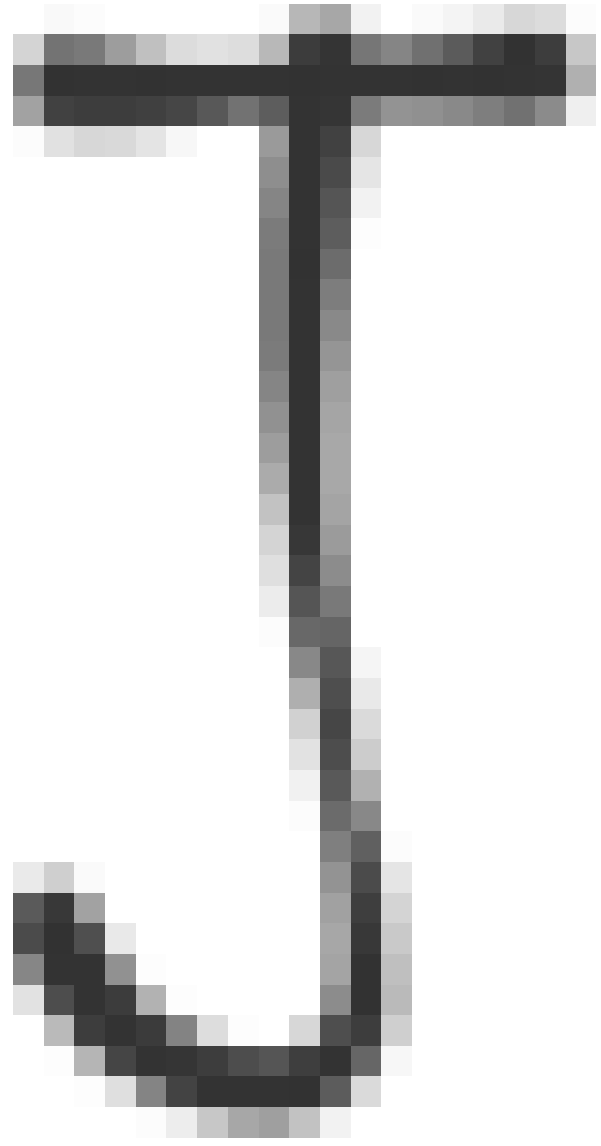


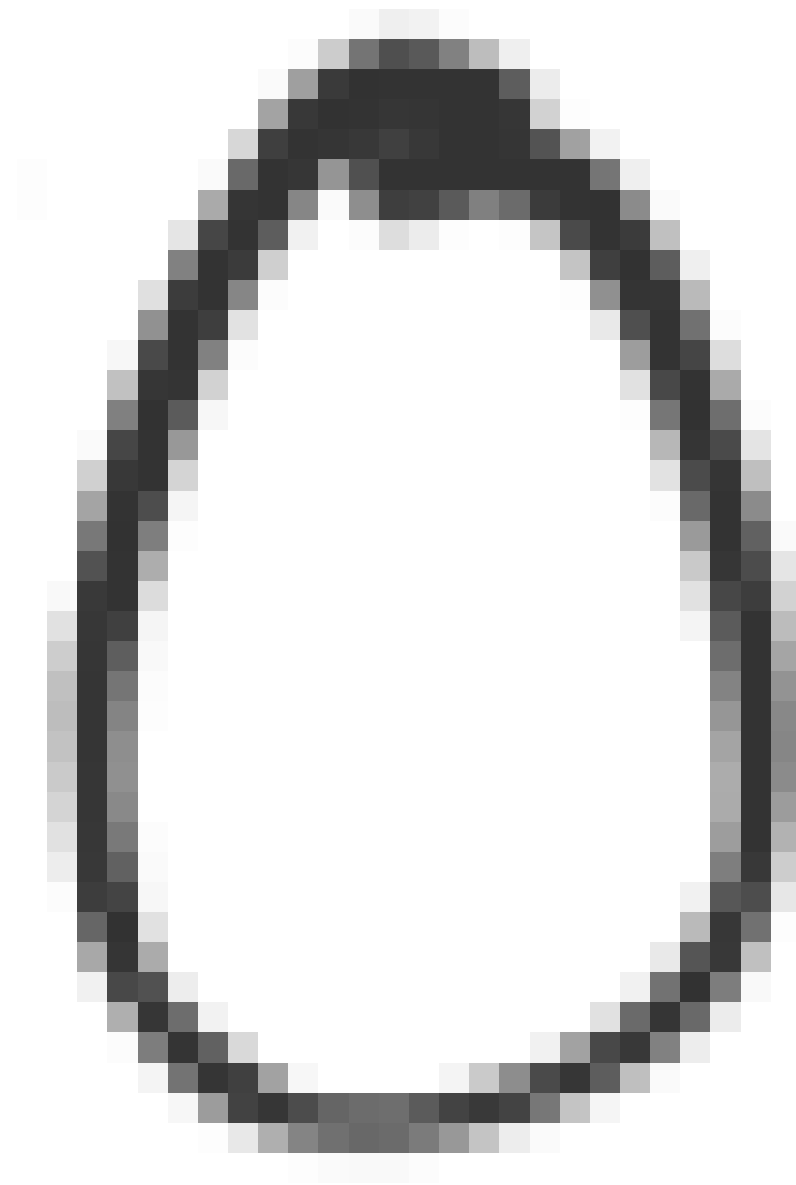


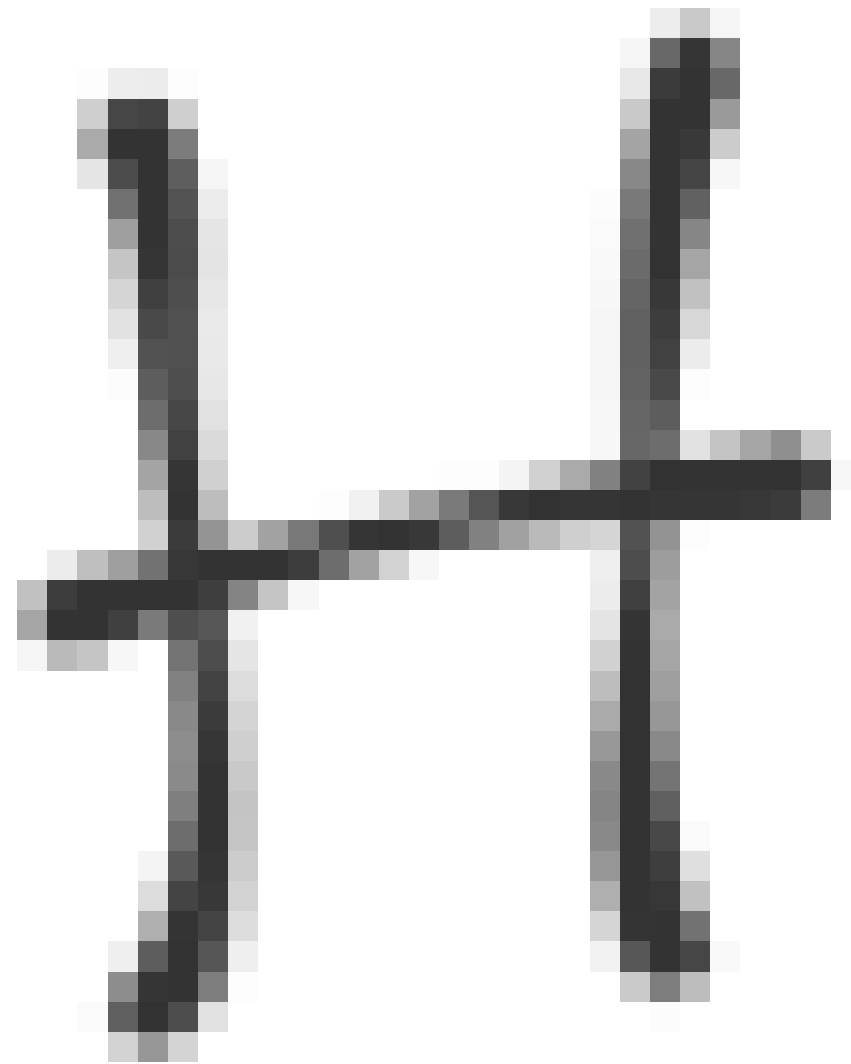




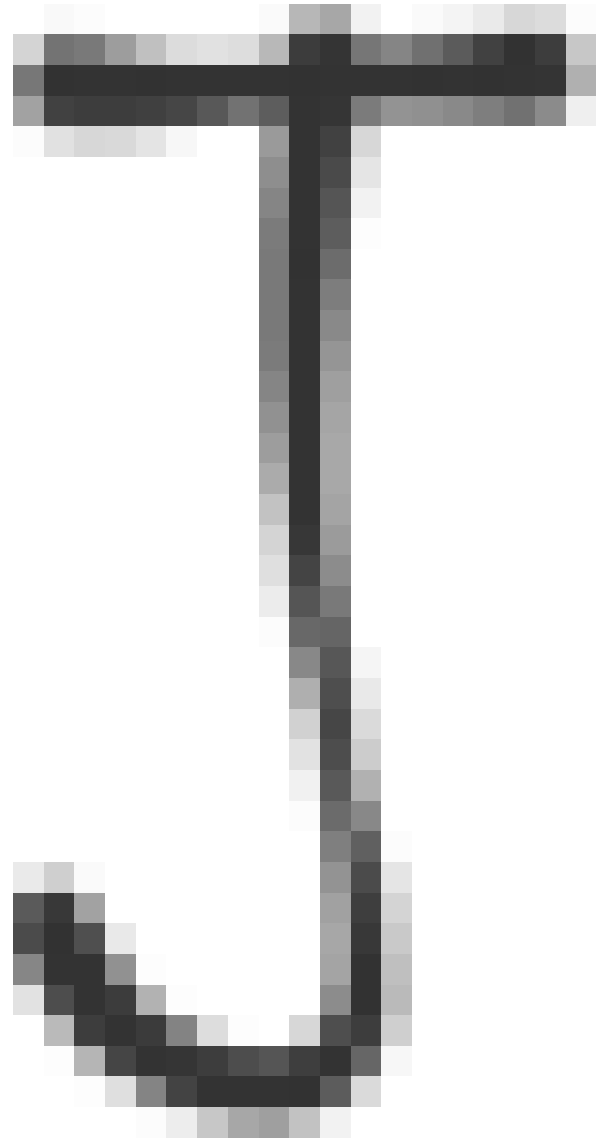
W

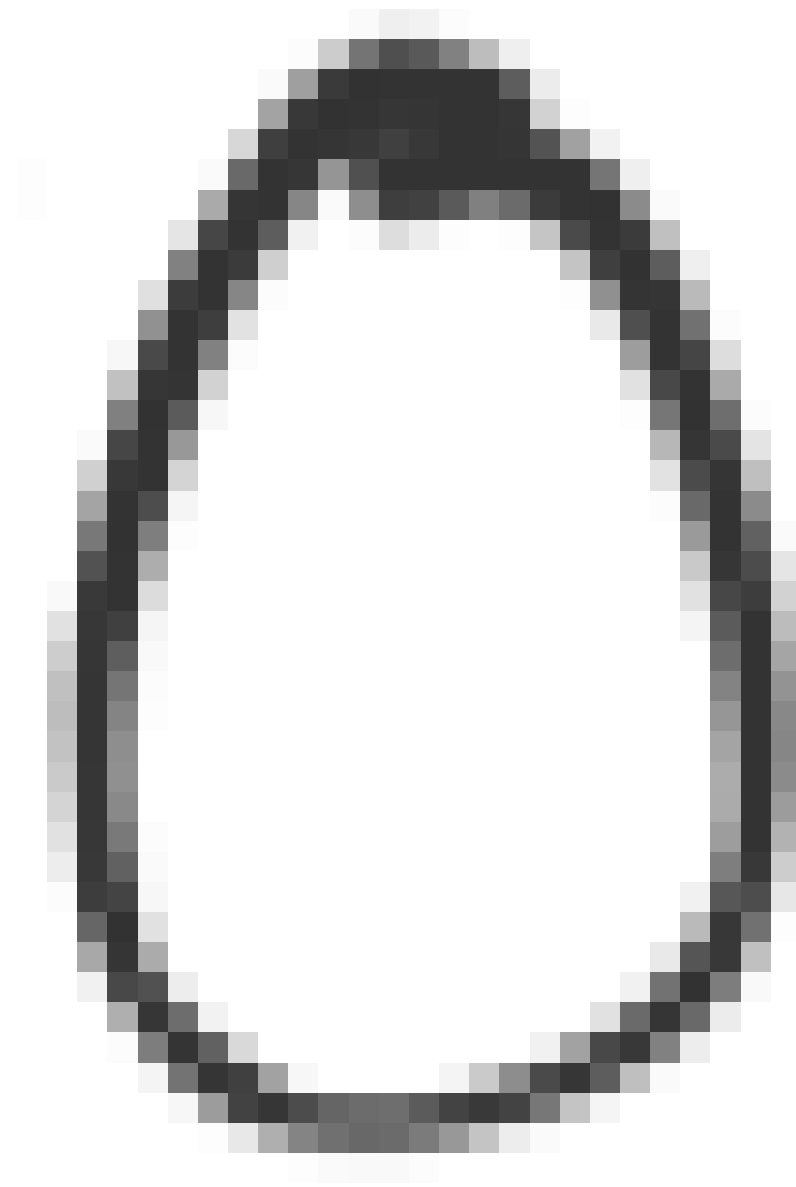




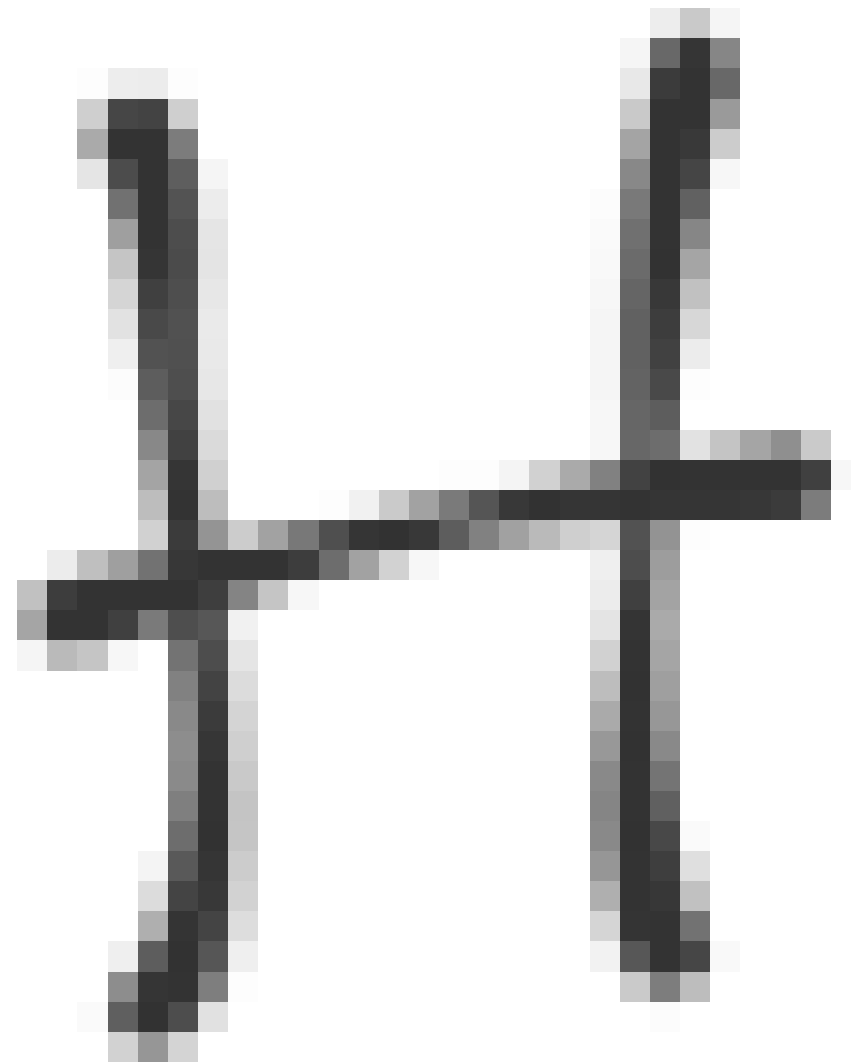


W



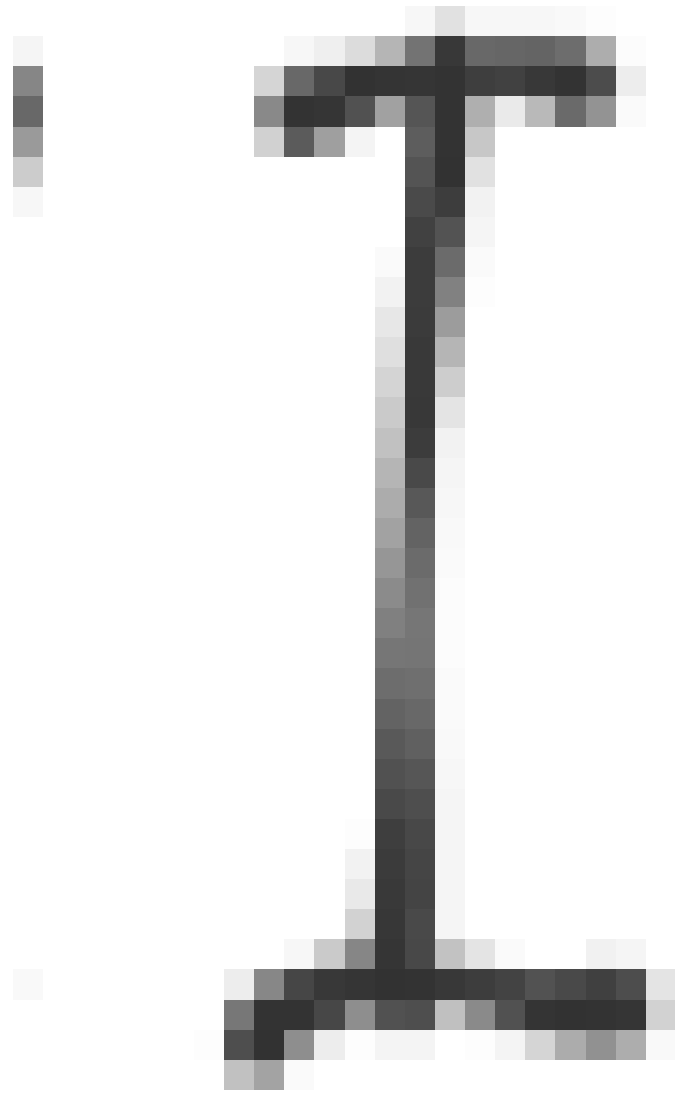


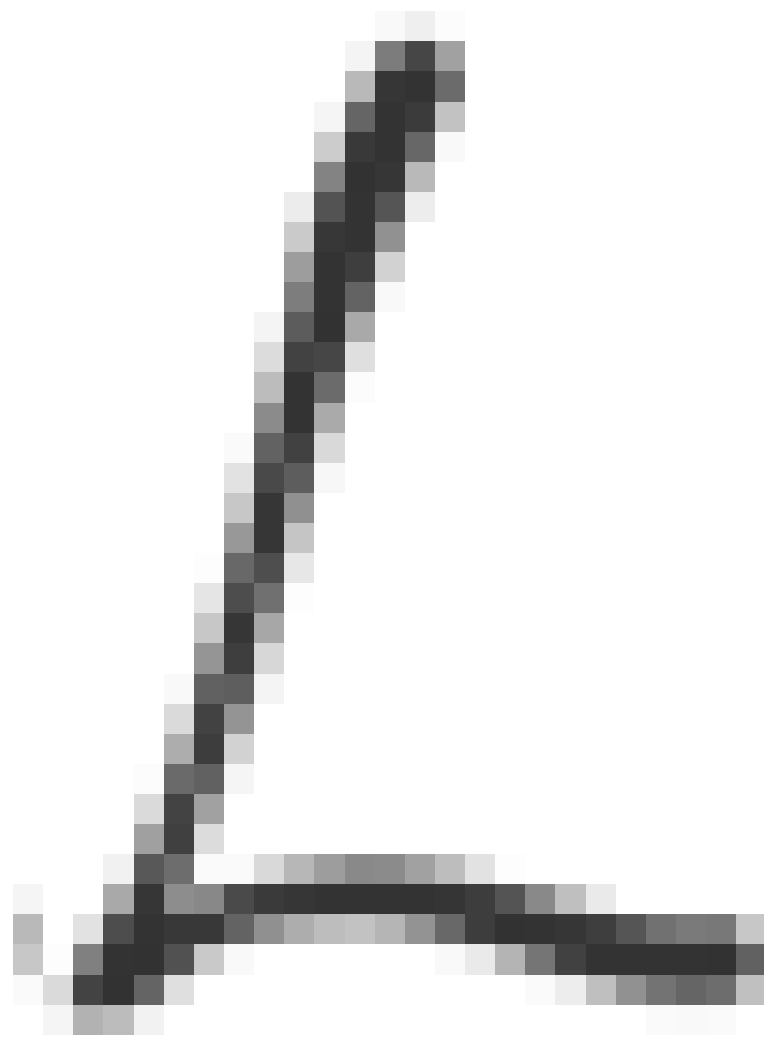


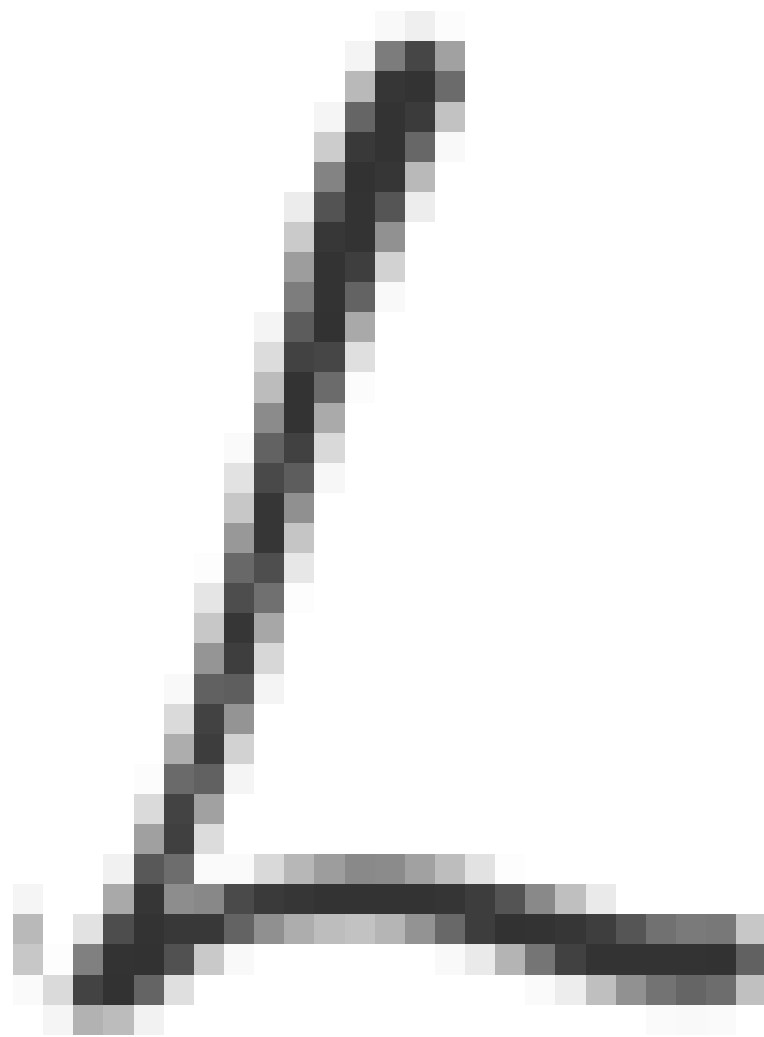


W

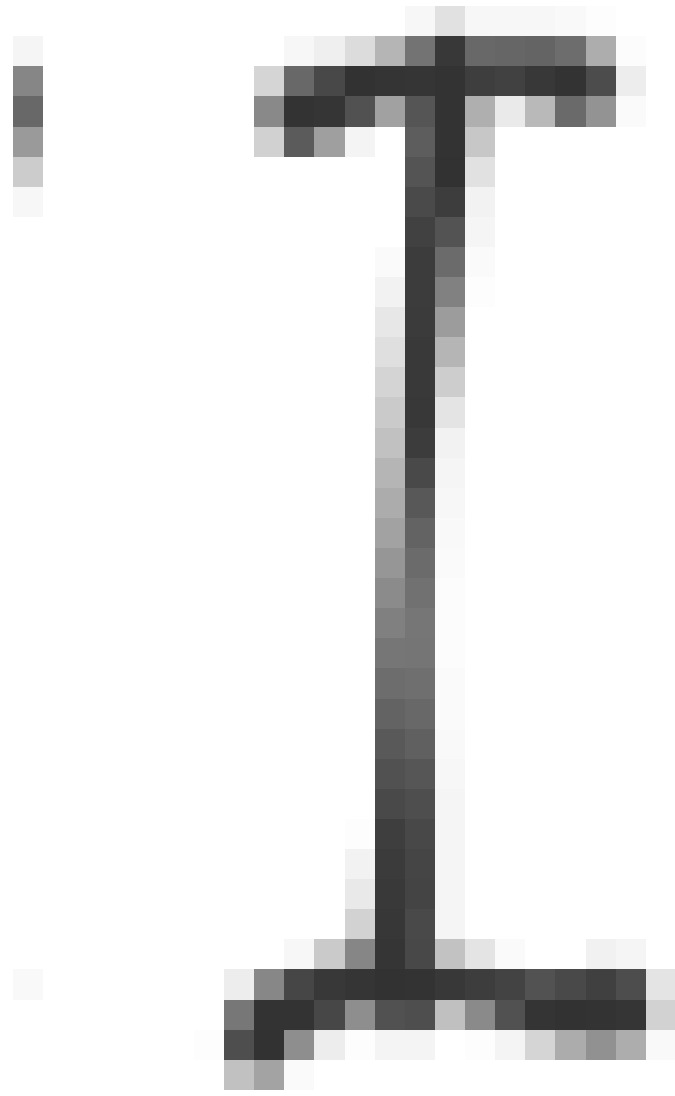




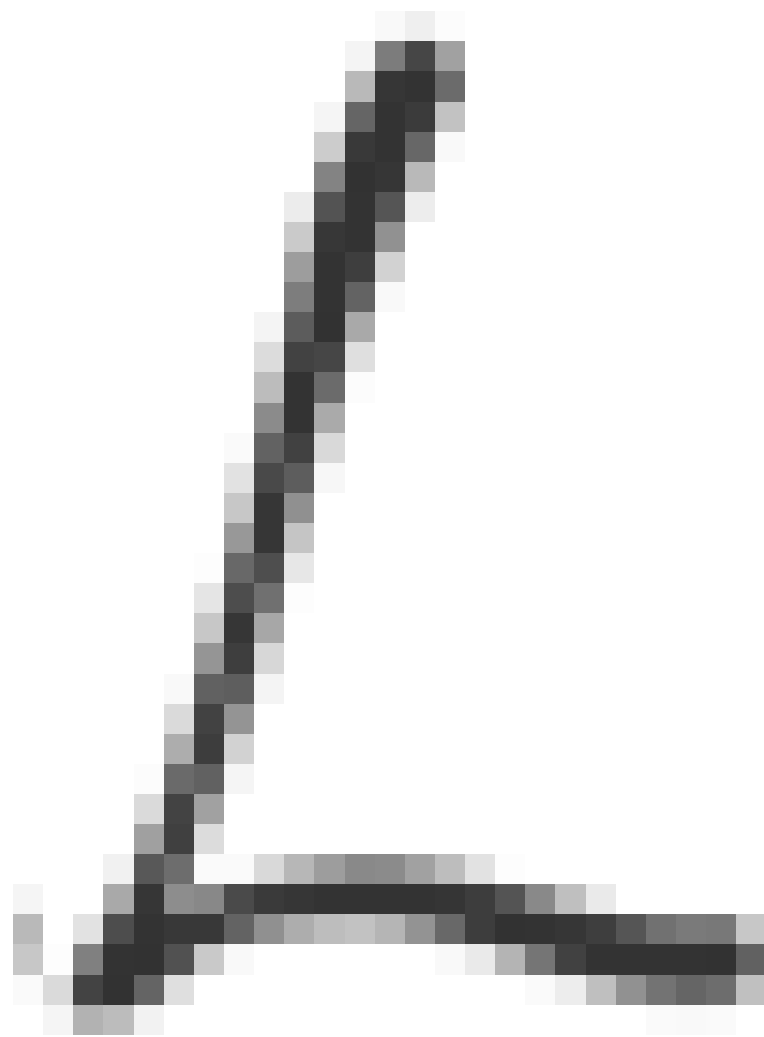


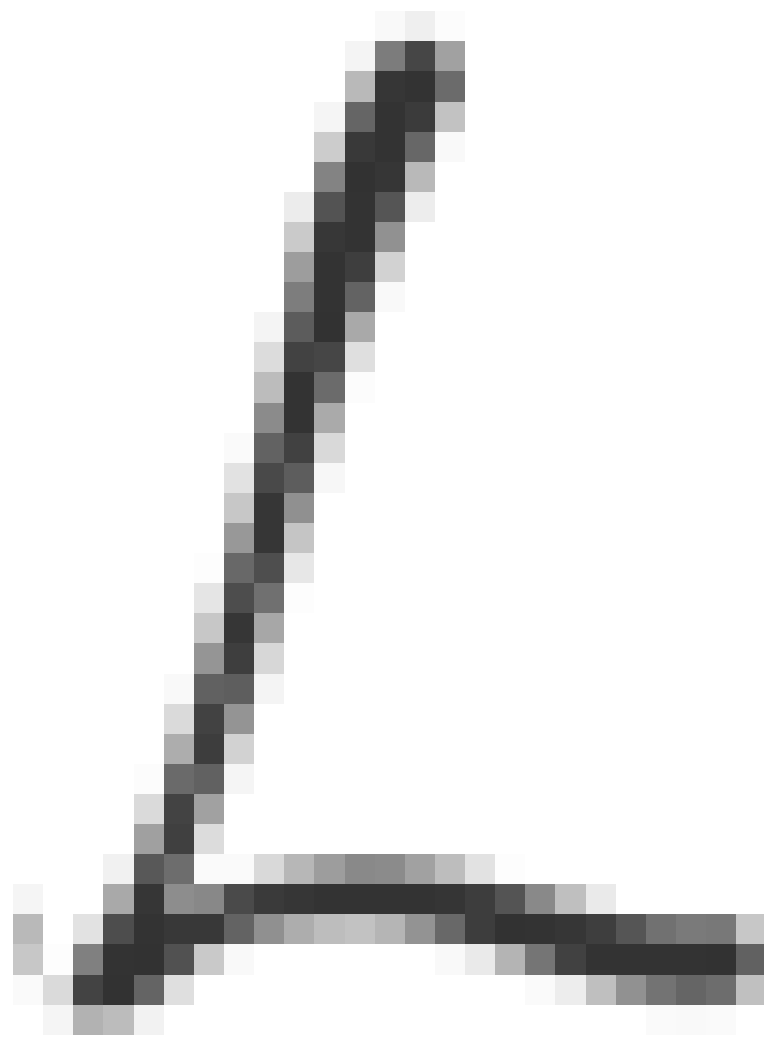




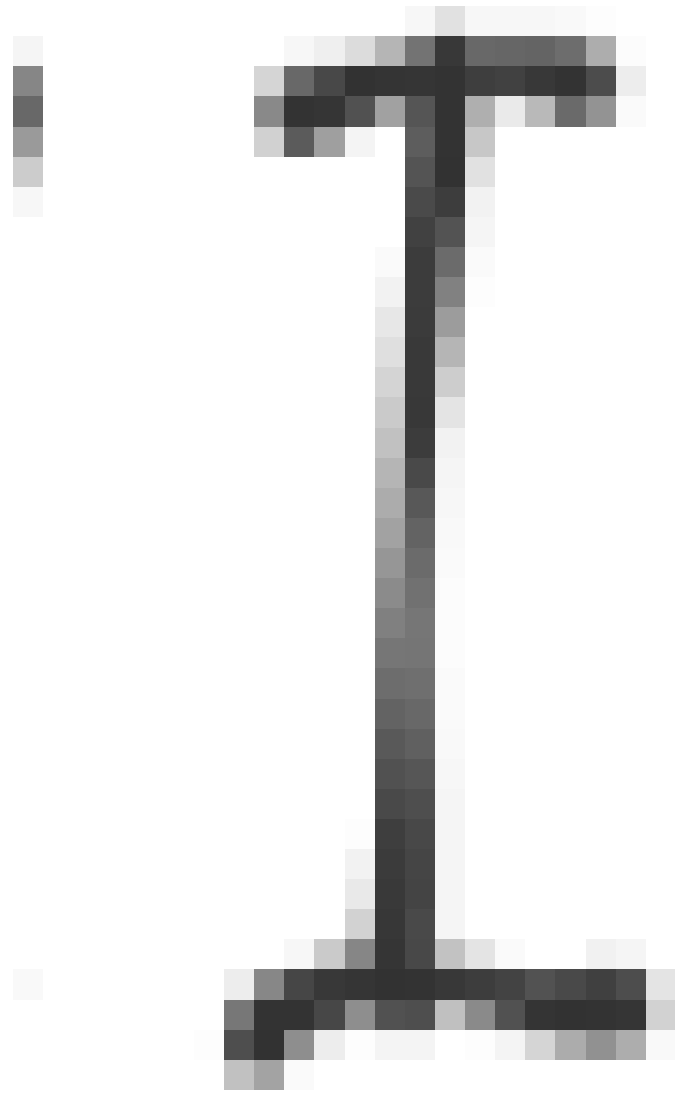


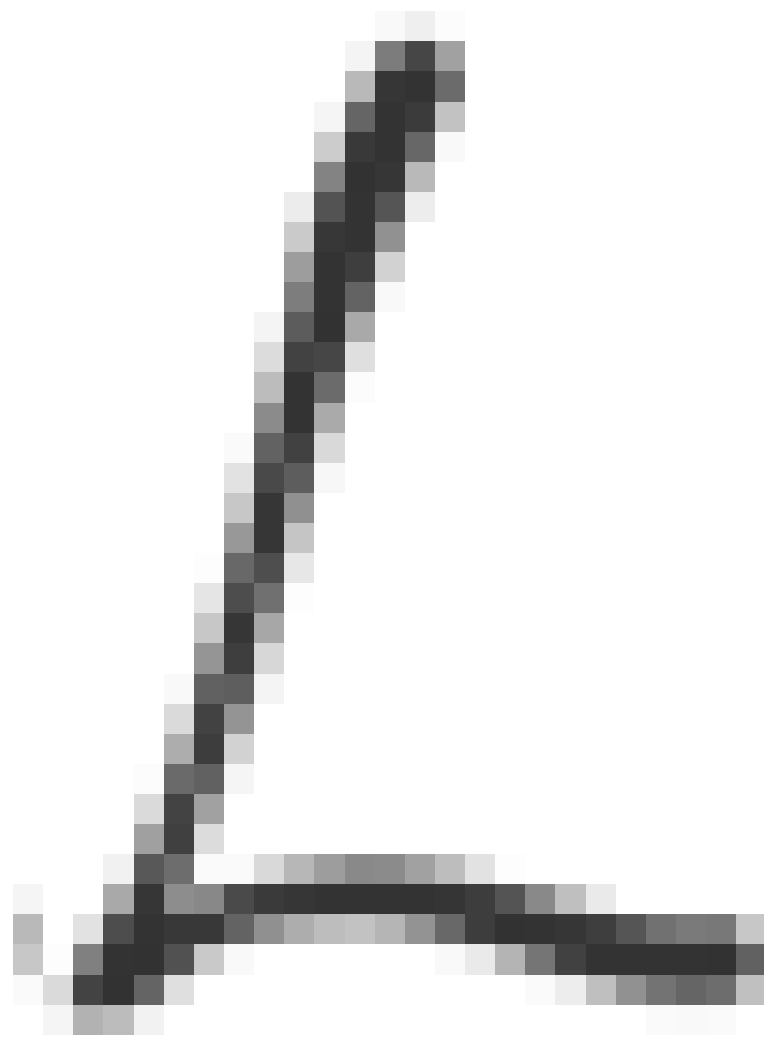


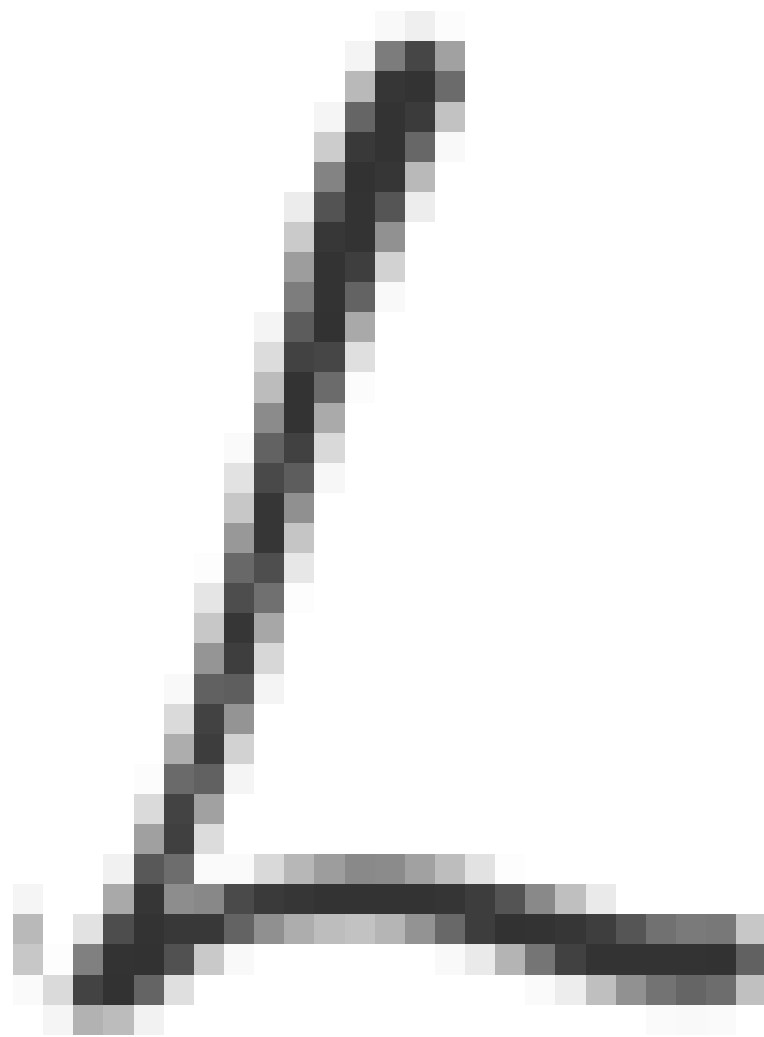




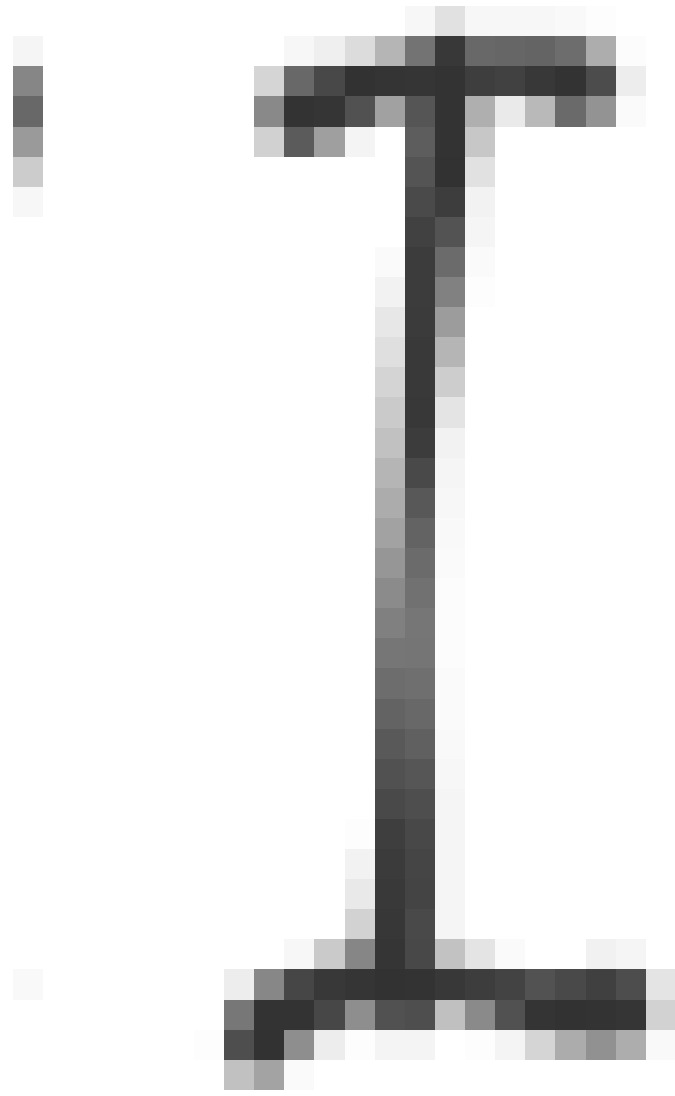




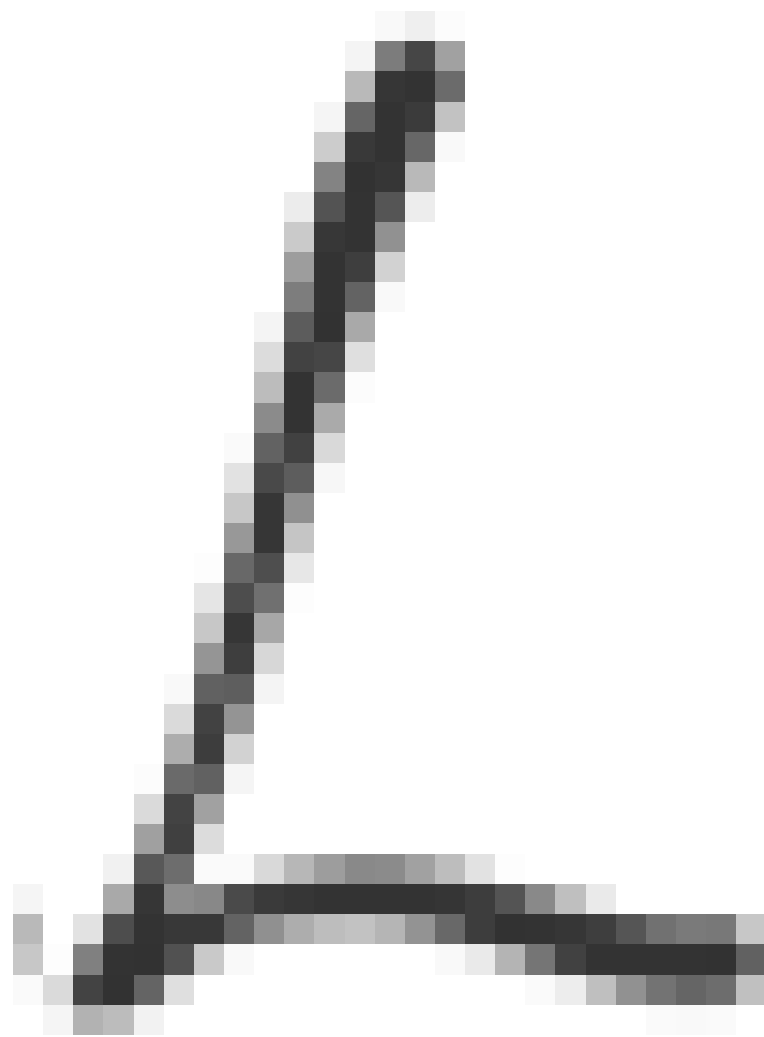


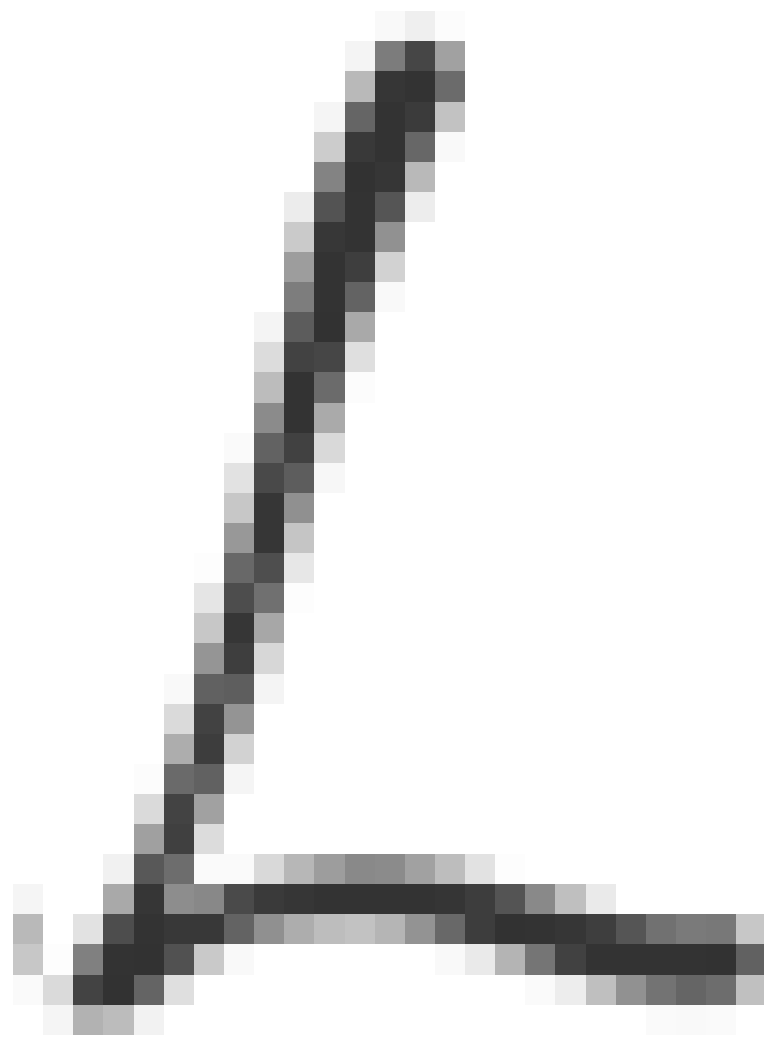




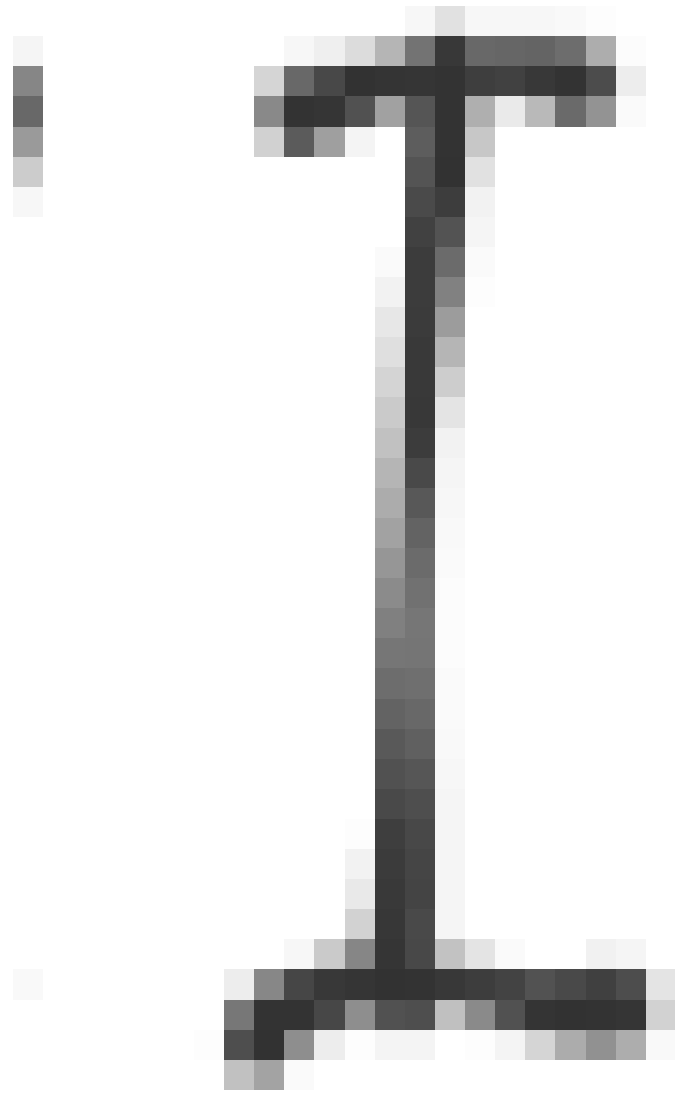


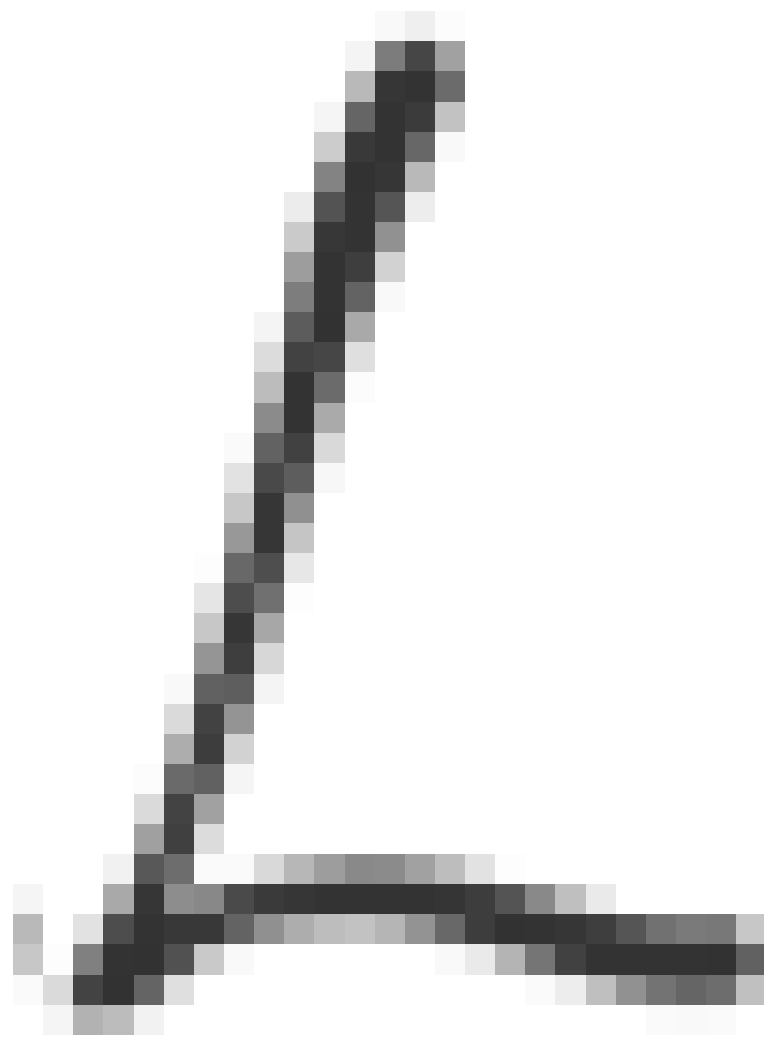


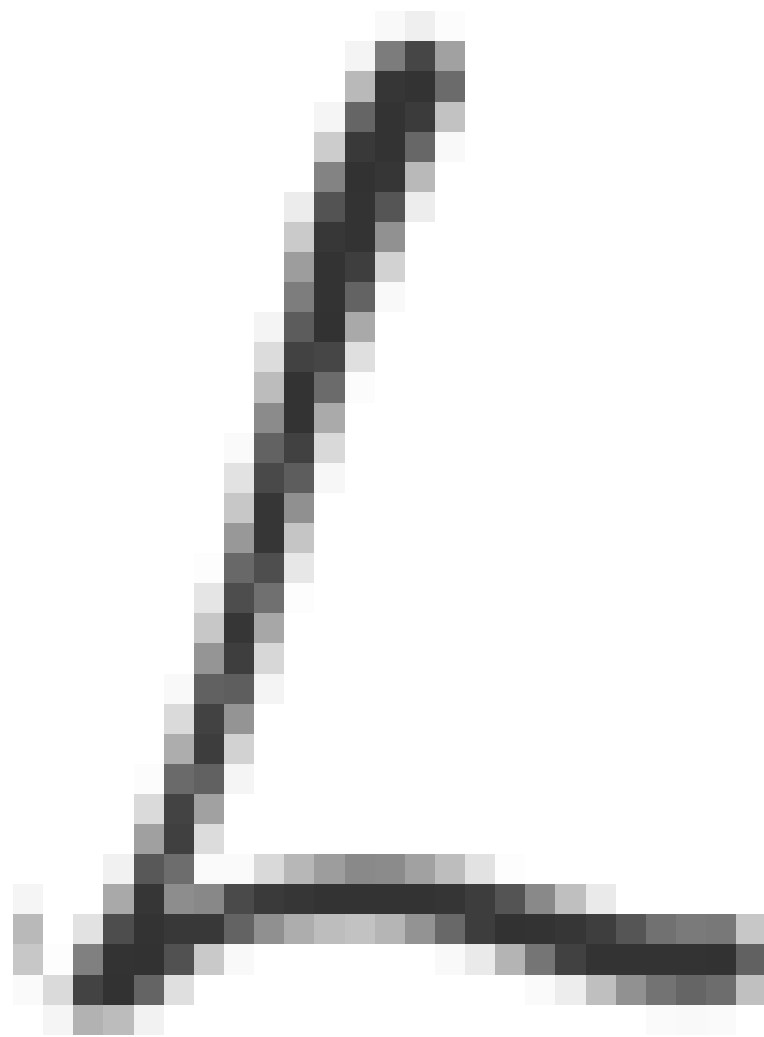


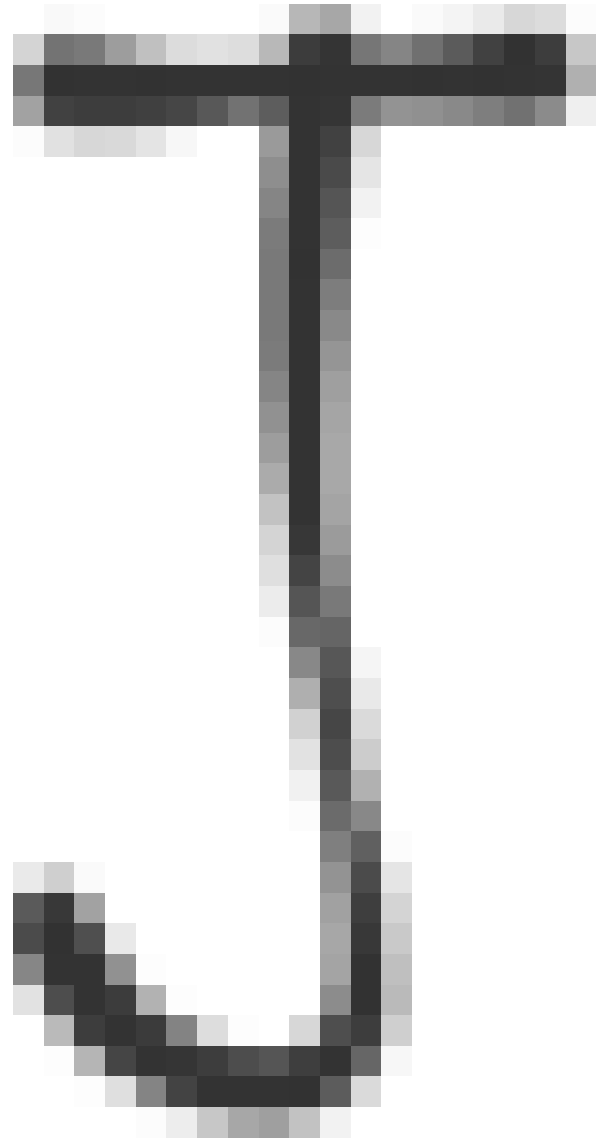


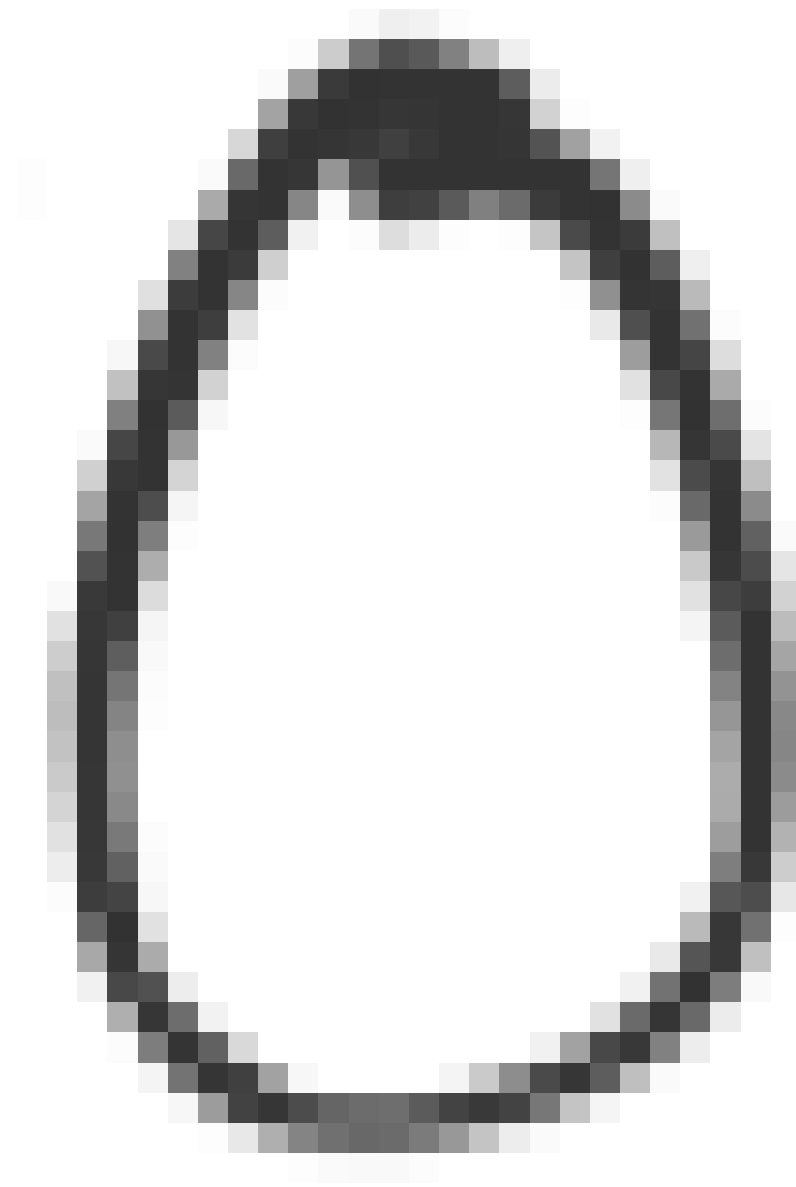




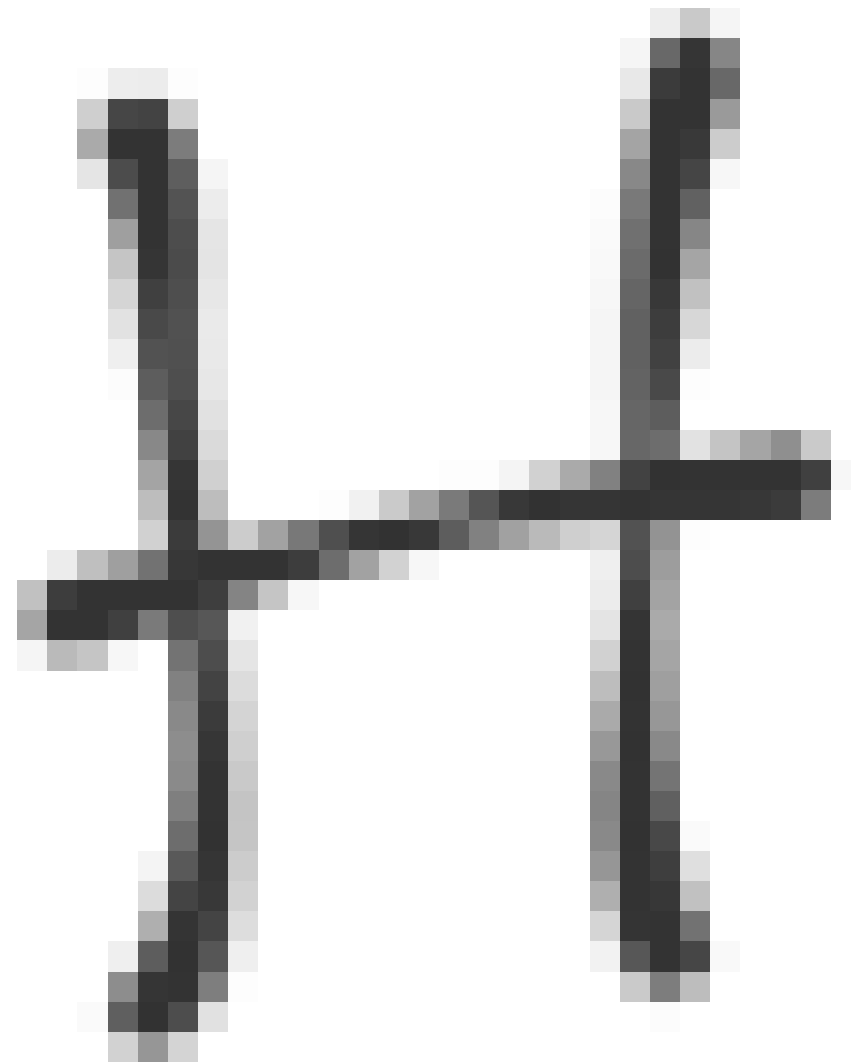












W