Содержание

1 Вывод формул

Для эффективной технической реализации необходимо применять 2 типа быстрых операций:

Кумулятивные суммы, на которых подробно не буду останавливаться

Свертки:

для каждого i, j вычислить

$$S(i,j) = \sum_{k1=0}^{K1} \sum_{k2=0}^{K2} X_{i+k1,j+k2} * mask_{k1,k2} \tag{1}$$

где X, mask - матрицы, а i, j пробегает все индексы при которых i + K1, j + K2 остаются внутри матрицы X.

Такую операцию можно представить как умножение на блочно тёплицеву матрицу с тёплицевыми блоками и после такого представления эффективно реализовать через fft (Fast Fourier Transform).

Получим необходимые для EM алгоритма формулы.

E-шаг:

$$q(d) = p(d|X,\theta,A) = \prod_{k=1}^{K} p(d_k|X_k,\theta,A) = \prod_{k=1}^{K} \frac{1}{C_k} p(X^k|d_k,\theta)p(d_k|A)$$

где q(d) - новое распределение, d_k - пара координат угла изображения (i_k, j_k), θ все параметры, A - prior для каждого d_k, C_k - обоснованность, относительно d_k - константа.

Так как распределение $p(d_k|X_k,\theta,A)$ дискретное по d_k, то константы можно не вычислять, так-как она просто норомирует сумму по всем d_k в единицу (а это можно просто проделать отдельной операцией нормирования).

Таким образом нас интересует следующее: для каждого (i, j) вычислить с точностью до константы

$$p(X^k|d_k,\theta)p(d_k|A), \text{ при } d_k = (i,j)$$

$$p(X^k, d_k|\theta, A) = p(X^k|d_k,\theta)p(d_k|A) =$$

$$((2\pi s^2)^{\frac{-HW}{2}})(\prod_{k1,k2 \in \text{face(i, j)}} \exp\frac{-1}{2s^2}(X^k_{k1,k2} - F_{k1-i,k2-j})^2)(\prod_{k1,k2 \notin \text{face(i, j)}} \exp\frac{-1}{2s^2}(X^k_{k1,k2} - B_{k1,k2})^2)A_{i,j}$$

где $d_k = (i, j)$, face(i, j) это область пикселей в которой находится лицо для данного d_k, вычисляемая как $[i : i + h - 1]x[j : j + w - 1]$, остальные k1, k2 проходят вдоль всей картинки. Теперь домножим и поделим на произведение экспонент разностей квадратов для фона картинки вдоль лица, тогда получим:

$$p(X^k, d_k|\theta, A) = ((2\pi s^2)^{\frac{-HW}{2}})(\prod_{k1,k2 \in \text{face(i, j)}} \exp \frac{-1}{2s^2}(X^k_{k1,k2} - F_{k1-i,k2-j})^2 - (X^k_{k1,k2} - B_{k1,k2})^2) * \\ *(\prod_{k1,k2} \exp \frac{-1}{2s^2}(X^k_{k1,k2} - B_{k1,k2})^2) A_{i,j} \tag{2}$$

Но тогда от i, j в формуле зависит только произведение вдоль лица и $A_{i,j}$, а остальное будет константой, а значит пропадет при нормировании вероятностей к единице.

$$p(d_k|X^k, \theta, A) \propto A_{i,j} \exp(logit(i, j)) \propto A_{i,j} \exp(logit(i, j) - M)$$

где $logit(i, j) = \frac{-1}{2s^2} \sum\limits_{k1,k2 \in face(i,j)} (X^k_{k1,k2} - F_{k1-i,k2-j})^2 - (X^k_{k1,k2} - B_{k1,k2})^2$
$M = max_{i,j}(logit(i, j))$

Таким образом задача свелась к подсчету логитов, ведь после получения пропорциональности, для получения самих вероятностей нужно всего лишь поделить их на сумму всех пропорциональных величин.

$$p(d_k|X^k, \theta, A) = \frac{A_{i,j} \exp(logit(i, j) - M)}{\sum\limits_{i,j} A_{i,j} \exp(logit(i, j) - M)}$$

Причем последняя формула вычислительно устойчива, так как все значения под экспонентами конечные числа меньше нуля, и происходит умножение на элементы матрицы A(сумма чисел в которой 1), а затем нормировка.

Логиты в свою очередь можно вычислять эффективно:

$$logit(i, j) = \frac{-1}{2s^2} \sum_{k1,k2 \in face(i,j)} (X^k_{k1,k2} - F_{k1-i,k2-j})^2 - (X^k_{k1,k2} - B_{k1,k2})^2 =$$

$$\frac{-1}{2s^2} \sum_{k1=0}^{h-1} \sum_{k2=0}^{w-1} (X^k_{k1+i,k2+j} - F_{k1,k2})^2 - (X^k_{k1+i,k2+j} - B_{k1+i,k2+j})^2 =$$

$$\frac{-1}{2s^2} \sum_{k1=0}^{h-1} \sum_{k2=0}^{w-1} (F_{k1,k2})^2 - 2(X^k_{k1+i,k2+j} F_{k1,k2}) + B_{k1+i,k2+j}(2X^k_{k1+i,k2+j} - B_{k1+i,k2+j})^2$$

Тогда первое слогаемое будет константой (его можно посчитать один раз), второе слогаемое

можно вычислить сразу для всех i, j по формуле свертки, так как оно представляется в виде формулы 1, а последнее слогаемое можно эффективно вычислить для каждого i, j за O(1) если сделать предподсчет 2d кумулятивной суммы последнего слогаемого.
Посчитав вероятности для каждого k, соберем их в матрицу q, это и будет ответом на E-шаге, а для map-версии алгоритма просто возьмем argmax по i,j для каждого k (это можно сделать еще до шага нормировки).

M-шаг:
Максимизируемый функционал выглядит так:

$$E_q[ln(p(X,d|\theta,A))] = \sum_{k=0}^{K} E_{q_k}[ln(p(X^k,d_k|\theta,A))] = \sum_{k=0}^{K}\sum_{i=0}^{H-h+1}\sum_{j=0}^{W-w+1} ln(p(X^k,d_k|\theta,A))q_k(i,j)$$

На самом деле это выражение суммирует только по тем i, j для которых $q_k(i,j) \neq 0$
Далее применив выражение 2 получим

$$E_q[ln(p(X,d|\theta,A))] = \sum_{k=0}^{K}\sum_{i=0}^{H-h+1}\sum_{j=0}^{W-w+1} q_k(i,j)(\frac{-HW}{2}ln(2\pi s^2)+ \tag{3}$$

$$\frac{-1}{2s^2}(\sum_{k1,k2\in\text{face(i, j)}}(X^k_{k1,k2}-F_{k1-i,k2-j})^2-(X^k_{k1,k2}-B_{k1,k2})^2)+$$

$$\frac{-1}{2s^2}(\sum_{k1,k2}(X^k_{k1,k2}-B_{k1,k2})^2)+ln(A_{i,j}))=$$

$$\frac{-HWK}{2}ln(2\pi s^2)+\frac{-1}{2s^2}\sum_{k=0}^{K}(\sum_{k1,k2}(X^k_{k1,k2}-B_{k1,k2})^2)+\sum_{i=0}^{H-h+1}\sum_{j=0}^{W-w+1}ln(A_{i,j})\sum_{k=0}^{K}q_k(i,j)+$$

$$\frac{-1}{2s^2}\sum_{k=0}^{K}\sum_{i=0}^{H-h+1}\sum_{j=0}^{W-w+1}q_k(i,j)\sum_{k1,k2\in\text{face(i, j)}}(X^k_{k1,k2}-F_{k1-i,k2-j})^2-(X^k_{k1,k2}-B_{k1,k2})^2$$

Теперь можно получать формулы для точечных оценок параметров.
Рассмотрим выражение что зависит только от A в задаче:

$$\sum_{i=0}^{H-h+1}\sum_{j=0}^{W-w+1}ln(A_{i,j})\sum_{k=0}^{K}q_k(i,j)$$

для него заранее известен ответ, так-как если это выражение поделить на K, то получим минус кросс-энтропию, тогда она максимальна при совпадающих распределениях, а значит:

$$A_{i,j}^{optimal} = \frac{\sum\limits_{k=0}^{K} q_k(i,j)}{K} \tag{4}$$

Вычисялется одинаково как в обычном EM, так и в map-EM, только во втором случае это эффективно реализовывается инициализацией нулями, и последовательным добавлением единиц по индексам i, j для кааждого k
Рассмотрев отельно пиксель лица $F_{k1,k2}$ и зависящее от него выражение:

$$\frac{-1}{2s^2}\sum_{k=0}^{K}\sum_{i=0}^{H-h+1}\sum_{j=0}^{W-w+1} q_k(i,j)(F_{k1,k2} - X^k_{i+k1,j+k2})^2$$

Приравняв градиент к нулю, получим решение:

$$F_{k1,k2}^{optimal} = \frac{\sum\limits_{k=0}^{K}\sum\limits_{i=0}^{H-h+1}\sum\limits_{j=0}^{W-w+1} q_k(i,j)X^k_{i+k1,j+k2}}{K}$$

Это выражение напрямую считается эффективно через выражение 1, а для map-EM его можно приобразить следующим образом:

$$F_{k1,k2}^{optimal} = \frac{\sum\limits_{k=0}^{K} X^k_{i_k+k1,j_k+k2}}{K}$$

так как для каждого k: $q_k(i,j) \neq 0$ только для одной пары: i_k, j_k, для которой q = 1

Выделить выражение зависящее от $B_{k1,k2}$ легче в изначальной форме (до выделения разности квадратов с фоном по всем пикселям), тогда $B_{k1,k2}$ будет встречаться только в тех суммах, где лицо не загараживает этот пиксель, тогда выражение принимает вид:

$$\frac{-1}{2s^2}\sum_{k=0}^{K}\sum_{(i,j\text{: таким, что k1, k2}\notin face(i,j))} q_k(i,j)(b_{k1,k2} - X^k_{k1,k2})^2$$

Приравняв градиент к нулю, нетрудно получить:

$$B_{k1,k2}^{optimal} = \frac{\sum_{k=0}^{K} X^k_{k1,k2}\sum_{i,j\text{: таким, что k1, k2}\notin face(i,j)} q_k(i,j)}{\sum_{k=0}^{K}\sum_{i,j\text{: таким, что k1, k2}\notin face(i,j)} q_k(i,j)}$$

В общем случае числитель и знаменатель вычисляется за O(1) с помощью 2d кумулятивных сумм, так i, j подходящие под условие того что пиксель внутри лица образуют прямоугольник: [max(0, k1 + 1 - h), min(k1, H - h)] x [max(0, k2 + 1 - w), min(k2, W - w)], а значит эффективно вычисляется через кумулятивную сумму, а вычислить сумму где это условие не выполняется можно просто вычтя из суммы по всем (i, j).

Тогда итоговая формула для общего EM:

$$B_{k1,k2}^{optimal} = \frac{\sum_{k=0}^{K} X_{k1,k2}^{k}(1 - \sum\limits_{i=max(0,k1+1-h)}^{min(k1,H-h)} \sum\limits_{j=max(0,k2+1-w)}^{min(k2,W-w)} q_k(i,j))}{\sum_{k=0}^{K}(1 - \sum\limits_{i=max(0,k1+1-h)}^{min(k1,H-h)} \sum\limits_{j=max(0,k2+1-w)}^{min(k2,W-w)} q_k(i,j))}$$

В то же время для map-EM можно сократить вид формулы, до

$$B_{k1,k2}^{optimal} = \frac{\sum_{k=0}^{K} X_{k1,k2}^{k} I[i_k, j_k\text{: такие, что k1, k2} \notin face(i,j)]}{\sum_{k=0}^{K} I[i_k, j_k\text{: такие, что k1, k2} \notin face(i,j)]}$$

где I - индикатор, i_k, j_k - обозначают то же, что и в формуле для пикселей фона.

Такая формула куда нагляднее реализцется в коде, ведь можно наоборот для каждого i_k, j_k вычитать из суммы слогаемое для тех k1, k2 $\in face(i_k, j_k)$, тогда формула реализуется вычислительно эффективно.

Наконец запишем часть формулы зависящую от s:

$$\frac{-KHW}{2} ln(s^2) - \frac{C}{2s^2}$$

где $C = \sum_{k=0}^{K} \sum_{k1,k2}(X_{k1,k2}^{k} - B_{k1,k2})^2 + \sum_{k=0}^{K} \sum_{i,j} q_k(i,j) \sum_{k1,k2 \in face(i,j)} (X_{k1,k2}^{k} - F_{k1-i,k2-j})^2 - (X_{k1,k2}^{k} - B_{k1,k2})^2$

Последняя разность квадратов уже вычислялась эффективно на E-шаге для всех (i, j), здесь повторим процедуру.

выражение выше имеет минимум для s^2:

$$s_{optimal}^2 = \frac{C}{KHW}$$

В map-EM появляется более эффективный способ посчитать C - брать в сумму только один элемент разности квадратов - i_k, j_k.

Наконец посчитаем нижнюю оценку лог-правдоподобия:

$$ELBO(q, \theta, A) = E_q[ln(p(X, d|\theta, A))] + H(q)$$

где первое слогаемое подробно расписано в формуле 3, а второе простая энтропия:

$$H(q) = -\sum_{k,i,j} q_k(i, j) ln(q_k(i, j))$$

Внутри первого слогаемого появляется вычисление С - определенного так-же как на М-шаге для вычисления s^2, его вычисляем эффективно через формулу 1, а остальное вычисляем обычным способом.

2 Эксперименты

Был сгенерирован датасет небольших картинок на котором и проводилось тестирование.

Генерация стратовых точек случайная от 0 до 1 для каждого пикселя.

Результат при использовании use_Map 2.3 - фон, 2.4 - лицо, для одной стартовой точки.

(В конце вышло что все алгоритмы достаточно хорошо фосстанавливают фон, потому далее будет обозреваться только лицо, а еще что численные значения ELBO хорошо сходятся с визуальным качеством картинки, потому буду предоставллять картинки и комментарии к ним.)

MAP=True для числа итераций=10 2.5 - уже лучше, но все еще над чем то стоит поработать.

Далее воспользуемся алгоритмом полного EM - restart=10, видно что качество заметно лучше 2.6

Так же я добавлял модификацию сглаживания A - на этапе вычисления M-шага для MAP=True сглаживать априорный параметр (иначе он сходится к нулю и не меняется , при использовании этого параметра в коде равным значение датасета / 10) смог получить хороший результат 2.7

Наконец настоящие фон 2.8 и приступник 2.9

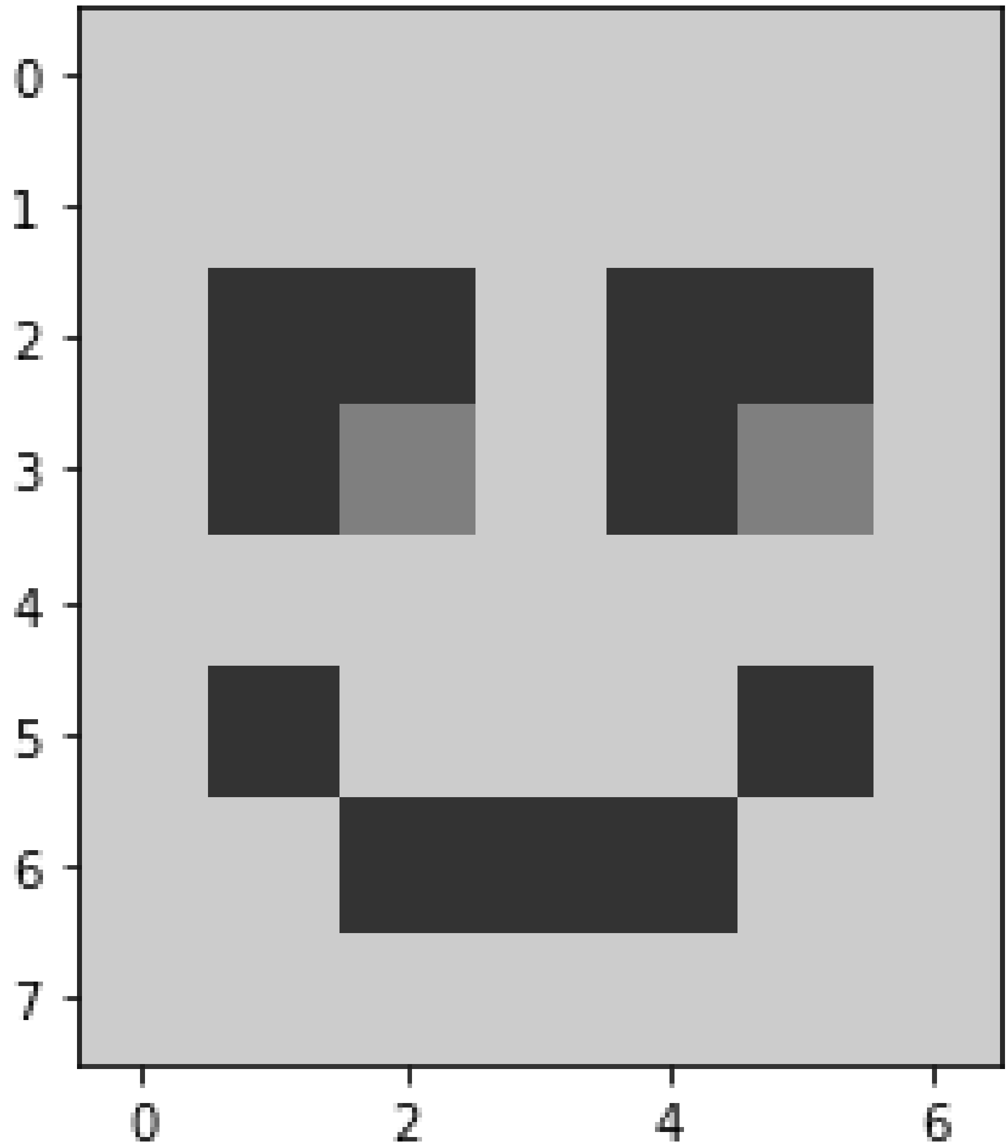

Рис. 2.1: Лицо

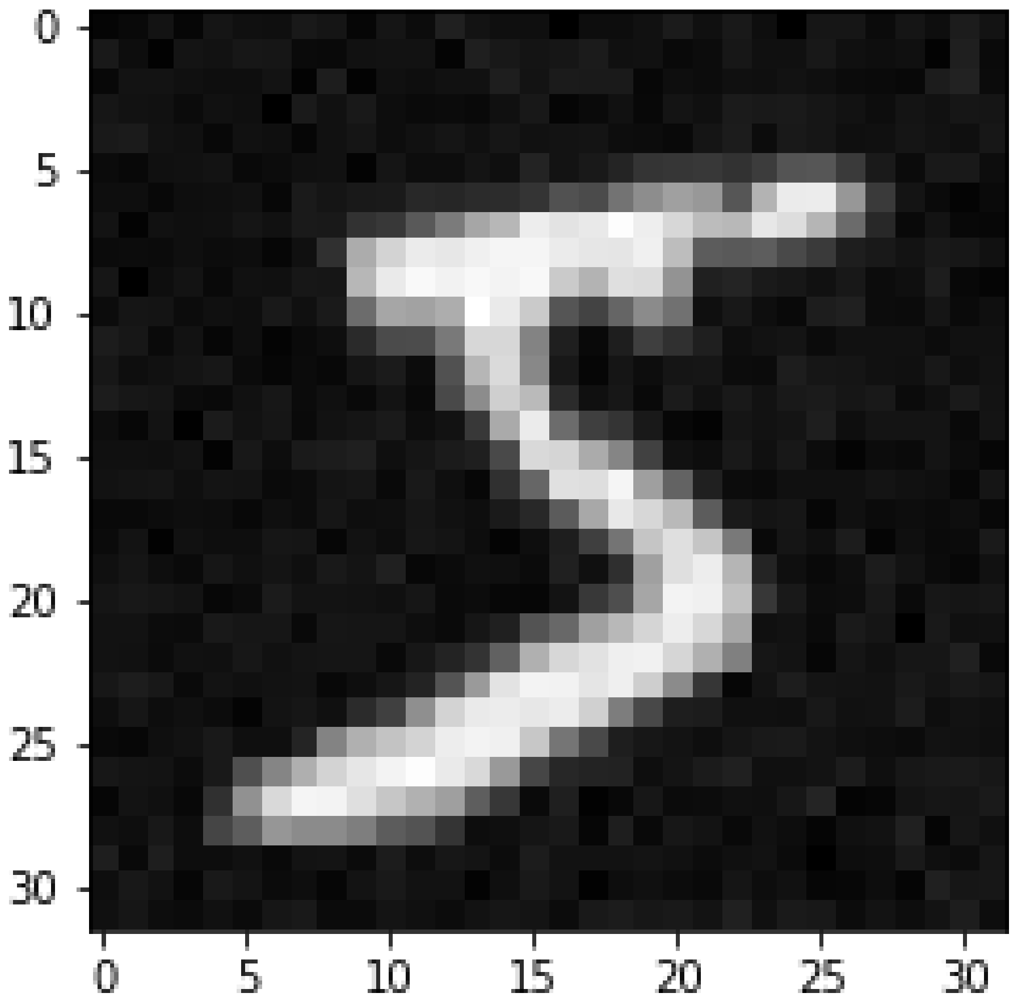

Рис. 2.2: Фон

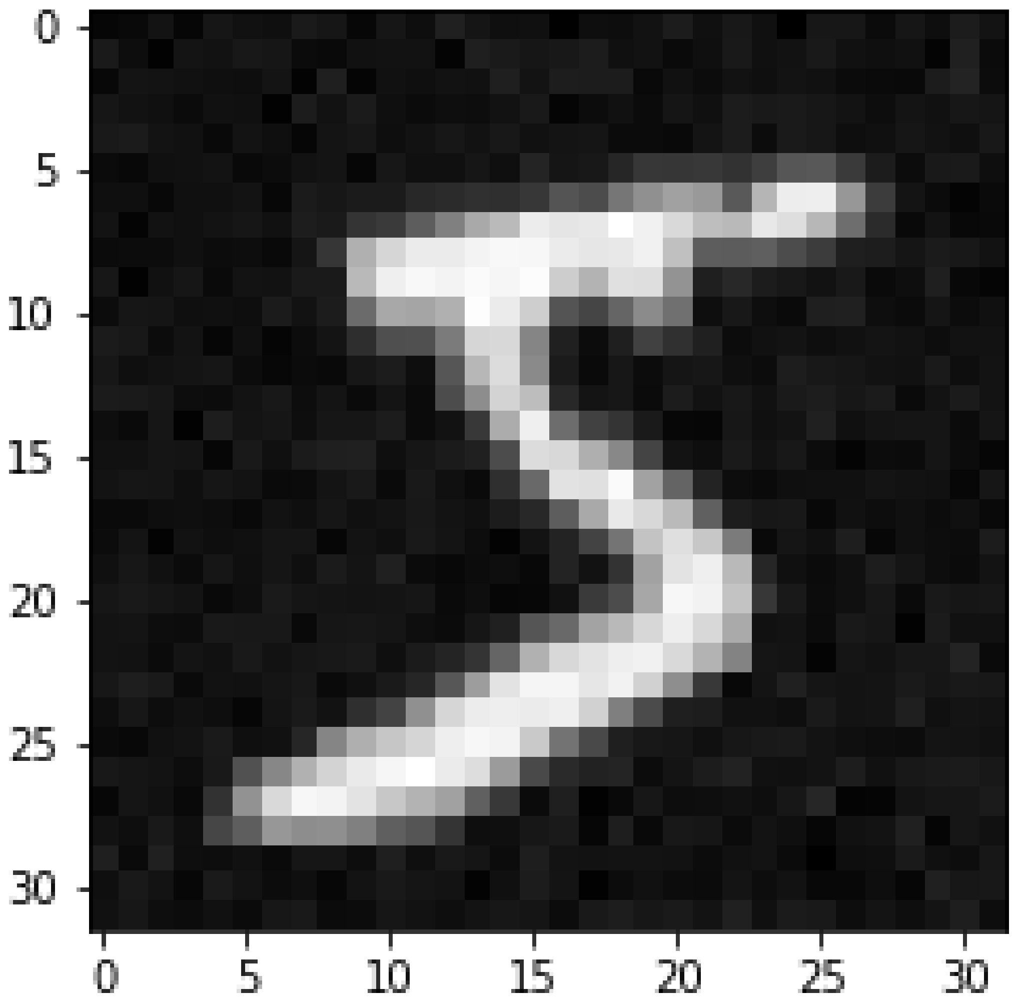

Рис. 2.3: Фон MAP=True

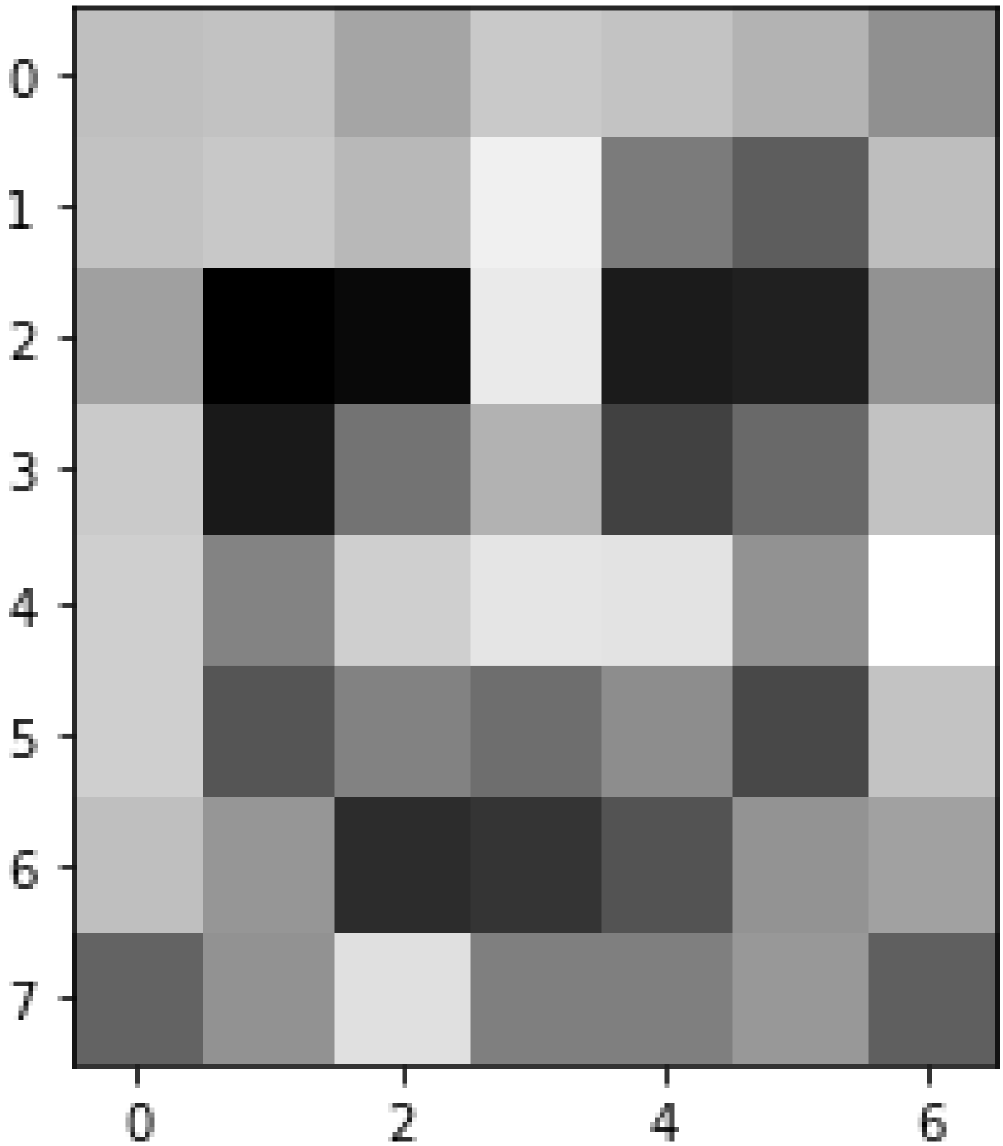

Рис. 2.4: Лицо MAP=True

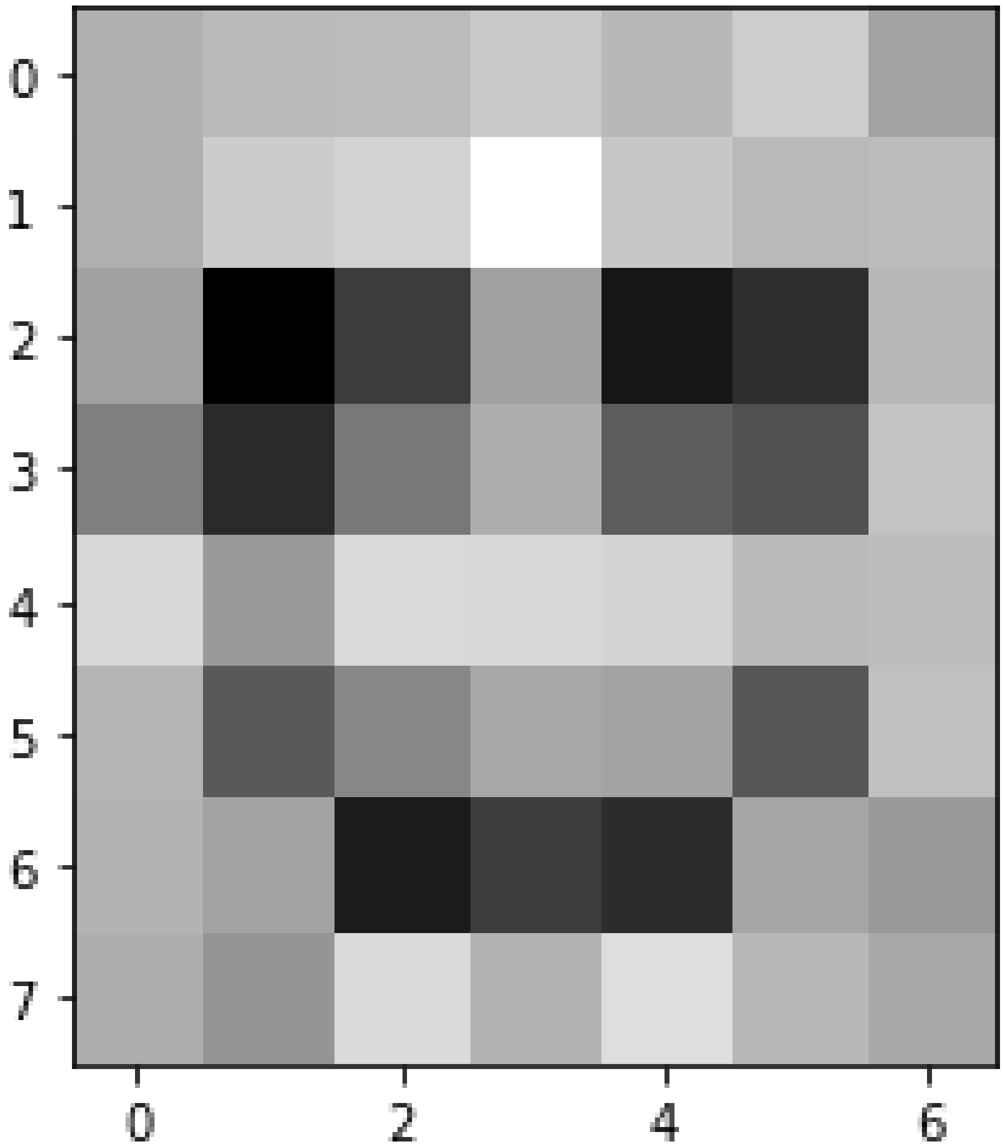

Рис. 2.5: Лицо MAP=True

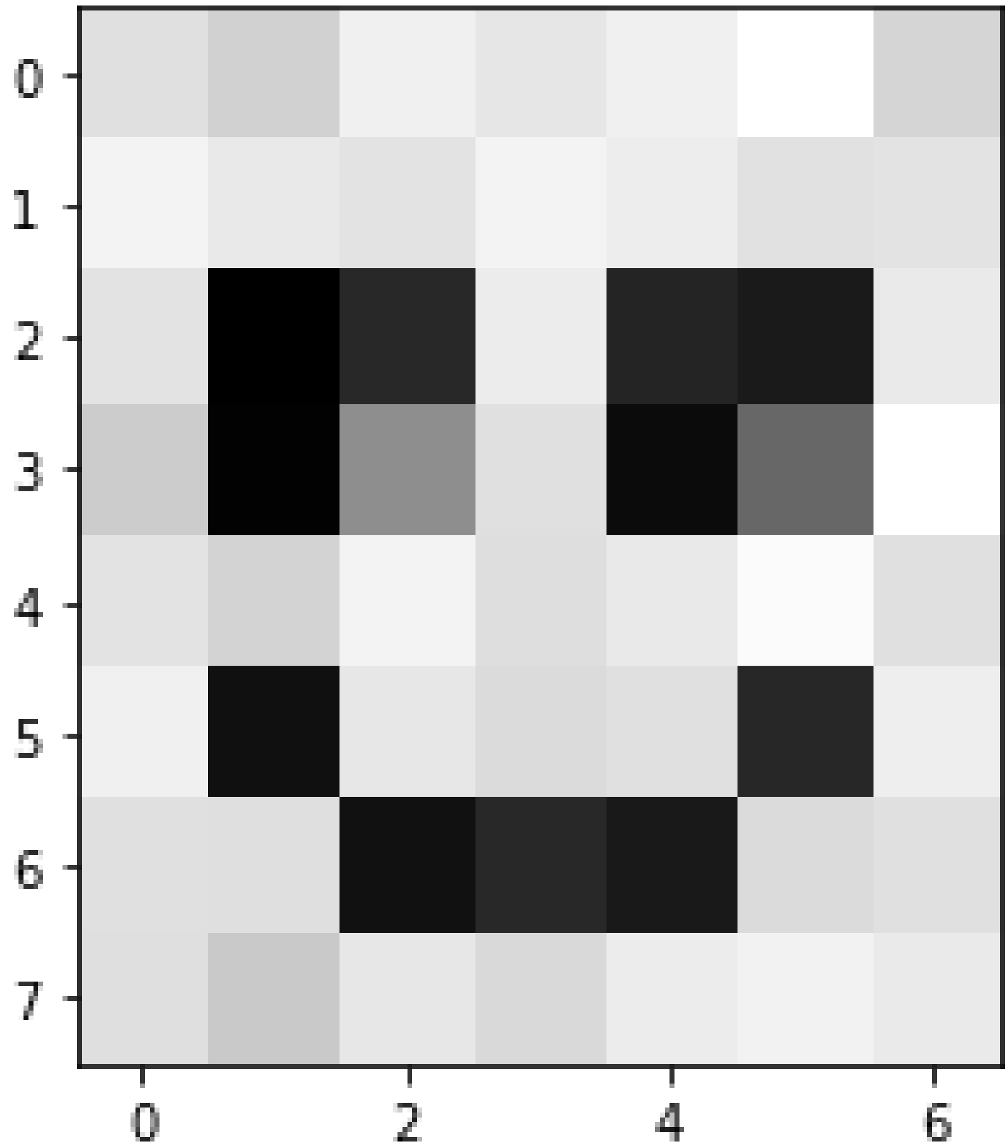

Рис. 2.6: Лицо

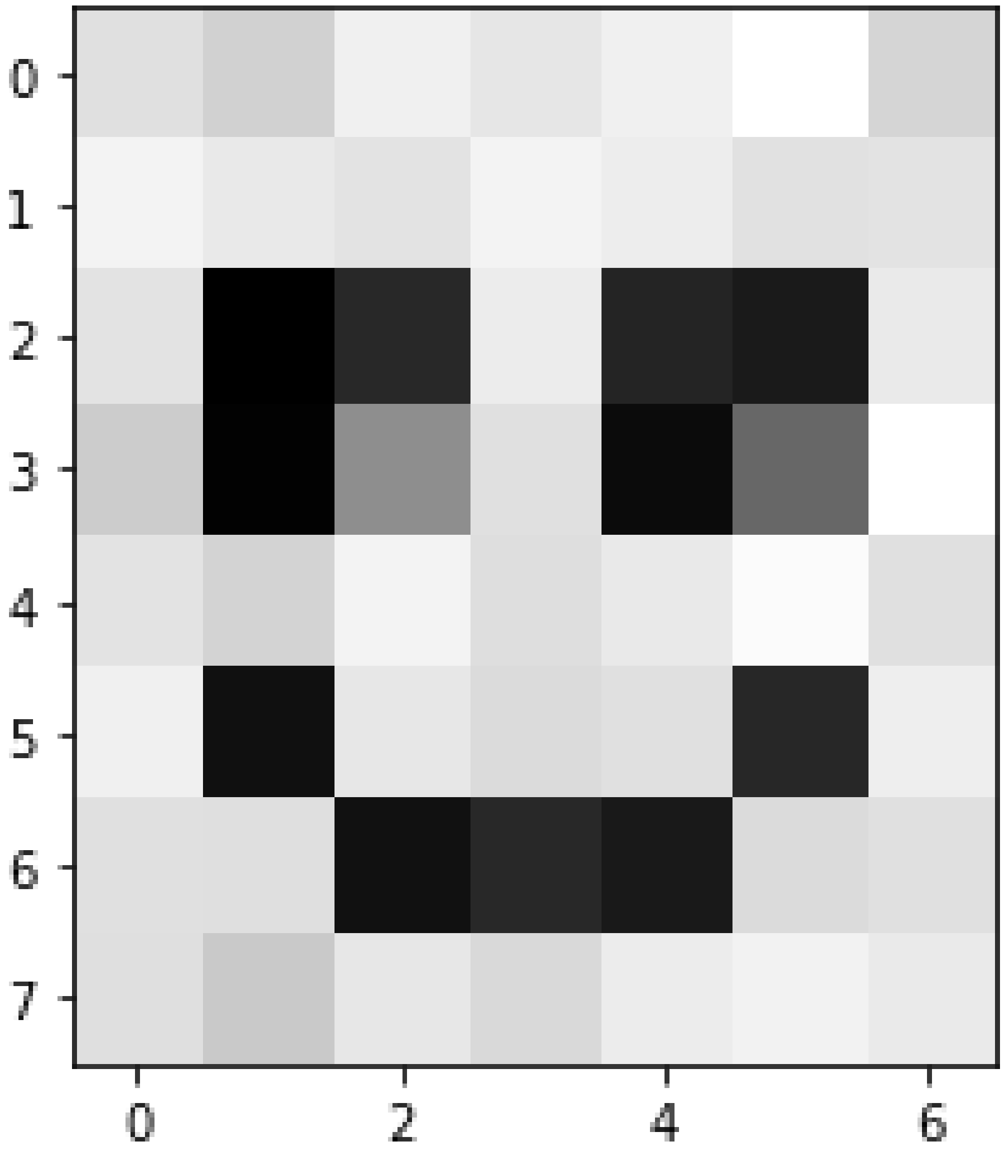

Рис. 2.7: Лицо prior_smoothing=100

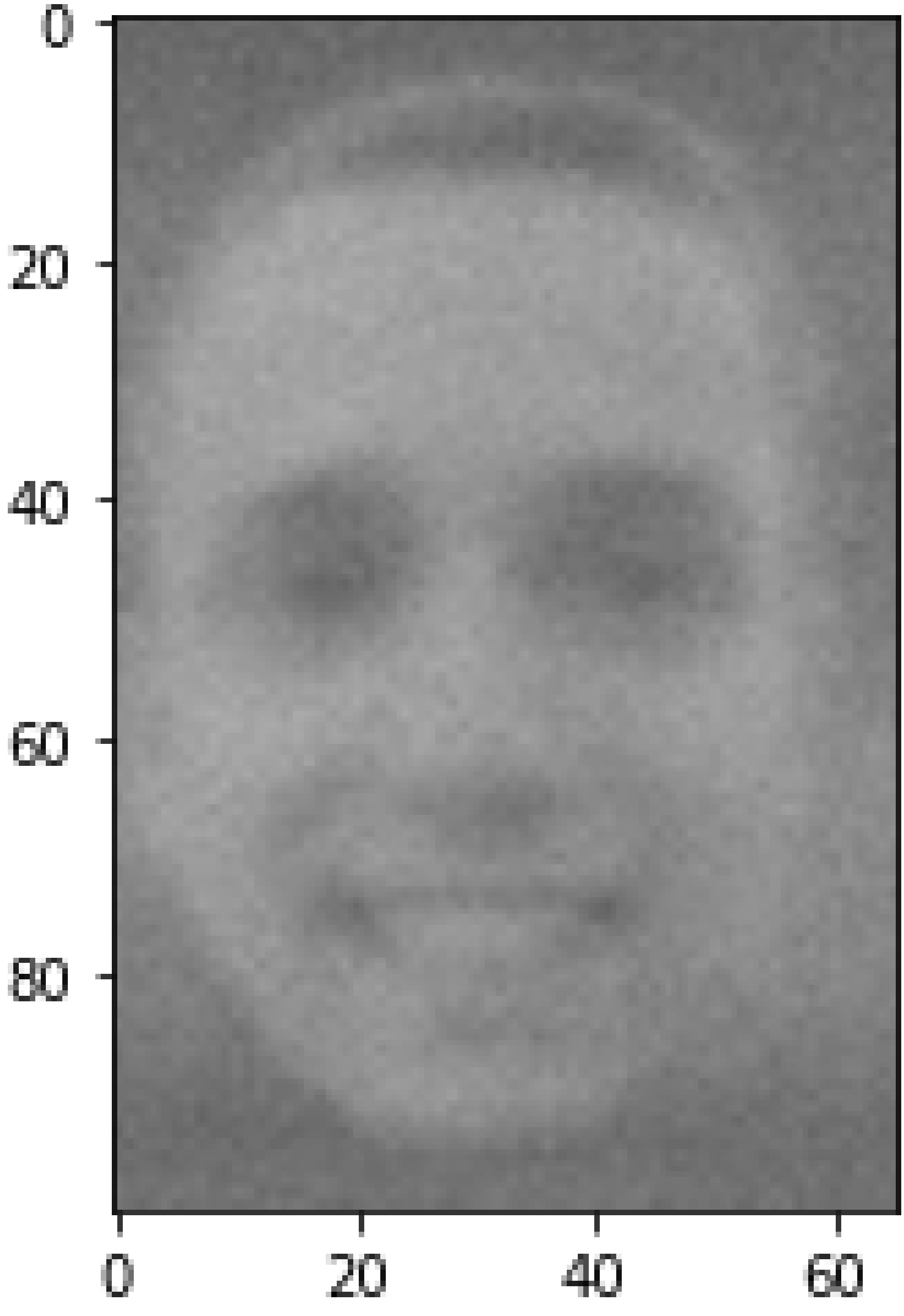

Рис. 2.8: Лицо

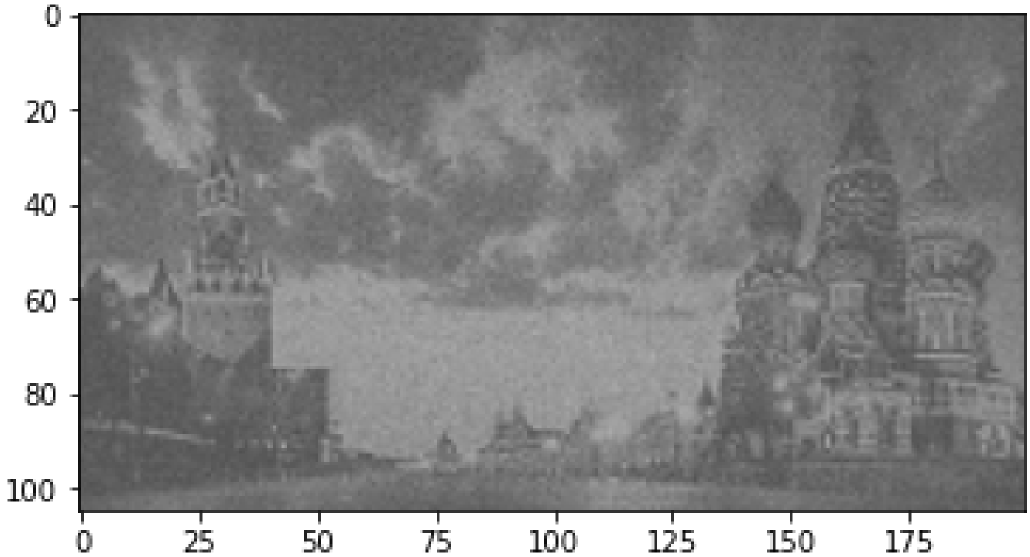

Рис. 2.9: Лицо