МАТЕМАТИЧЕСКИ МОДЕЛ НА КОНВОЛЮЦИОННИ НЕВРОННИ МРЕЖИ

Конволюционните невронни мрежи са изкуствени невронни мрежи с право разпространение на сигнала, които имат допълнителен слой, наречен конволюционен, заедно с редица други иновации, усъвършенствали идеите въведени от Неокогнитрона на Фукушима. Всяка невронна мрежа, която съдържа поне един конволюционен слой, може да се нарече конволюционна мрежа. Конволюционните мрежи са лидери в разпознаването на изображения и компютърното зрение. Имат многообразни приложения като откриване и етикетиране на обекти и хора в изображения; описване на изображения и видеоклипове с естествен език; проследяване на пътища и заобикаляне на препятствия с автономни автомобили и редица др.

Конволюционните невронни мрежи използват математическа операция конволюция, която е вид линейна операция, вместо матрично умножение, операция „обединяване“ (pooling) и извършват споделяне на параметри.

В конволюционните слоеве се извършва операция конволюция

$$f(t) = (x * w)(t) ,$$

която обикновено се означава с *, и обединява два набора от данни (информация). В конкретния пример, първият аргумент - функция **x**, с който се означават входните величини (input) и втория аргумент – тегловата функция **w**, която може да се представи като ядро или филтър (kernel). Резултатът от операцията конволюция се нарича карта на характеристиките (feature map). Приложената операция конволюция върху две функции произвеждат като резултат трета функция, която изразява как формата на едната се променя от другата. Филтърните матрици могат да се представят като детектор на характеристиките (feature detector), които се опитват да открият наличието на дадена характеристика от изображението.

Входният слой при конволюционните невронни мрежи съдържа променливите (пиксели на входното изображение), които могат да бъдат наблюдавани. Входният слой е последван от множество скрити слоеве, като в първия скрит слой на мрежата се извличат характеристики от ниско ниво - вертикални или хоризонтални ръбове, чрез сравняване на яркостта на съседните пиксели (фиг. 1). В последващите конволюционни слоеве, предвид описанието на ръбовете на първия скрит слой, се извличат ъгли, линии и контури, които са колекция от първоначално извлечените ръбове. На следващо ниво се откриват части от обекти, които съставляват колекция от линии и ъгли. В последния скрит слой, колекцията от извлечени части от изображението, се обединят в обект, който подлежи на разпознаване. В конволюционните мрежи се използват многомерни масиви – тензори. Всеки елемент от входните величини е многомерен масив от данни, а ядрото – многомерен масив от параметри. Тензорите, като елементи на входните величини и ядрото, се съхраняват поотделно, като се пазят стойности само за крайния набор от точки (2.48). Тензорите са масив от числа, подредени върху правилна мрежа с променлив брой оси. При използване конволюция между двумерно изображение имащо дискретен характер $I(j, j)$, и ядро K , функцията може да се представи съгласно следният израз:

$$f(i, j) = (I * K)(i, j) = \sum m \sum n\, I(m, n)K(i - m, j - n)$$

който израз може да се запише и еквивалентно:

$$f(i, j) = (K * I)(i, j) = \sum m \sum n\, I(i - m, j - n)K(m, n)$$

Операцията конволюция, често се заменя с близка до нея операция - кръстосана корелация, която е същата като конволюция, но без да се обръща ядрото:

$$f(i, j) = (I * K)(i, j) = \sum m \sum n\, I(i + m, j + n)K(m, n)$$

Единичният тензор може да се изрази посредством израза:

$$u_{i,j}^{(l-1)} = \sum_{m,n} w_{i,j,m,n}^{(l-1)} x_{i-m,j-n}^{(l-2)}$$

където:

- $u_{i,j}^{(l-2)}$- елемент (тензор) с координати (i,j);
- $w_{i,j,m,n}^{(l-2)}$ – многомерен масив от параметри – теглова матрица, съставляваща филтърна матрица или т. нар. филтърна маска (filter mask);
- $x_{i-m,j-n}^{(l-1)}$ – многомерен масив от входни величини, или многомерен масив от величини от предходно ниво.

В конволюционните слоеве, детекторите на характеристиките са конволюционни филтри, които обхождат изображението и изпълняват сумираща операция с претеглена сума. Всеки от тези филтри генерира карта на отговорите или т. нар. карта на характеристиките. Дълбочината d съответства на броя филтри, които се прилагат върху изображението в конволюционния слой. При цветни изображения, в първия конволюционен слой,$d = 3$, съответстващ на 3 цветови канала (RGB), като се прилагат 3 филтъра за всеки цвят по отделно. Всеки един пиксел от тези карти на характеристиките може да се разглежда като един неврон за следващия слой. Конволюционните слоеве имат няколко хиперпараметъра, които трябва да бъдат зададени предварително – размер на филтрите, брой филтри, отместването е стъпката (stride), с която избраната филтърна матрица обхожда входната матрица и допълване или разширяване на картата на характеристиките (padding). Операцията конволюция е прилагане на филтри към вход, което води до активиране на неврон. Многократното прилагане на един и същи филтър към вход, води до карта на активациите, наречена карта на характеристиките, указваща местоположенията и силата на открита характеристика във изображение. Филтърните маски могат да научават и индивидуални филтри за откриване на ръбове, въз основа на вероятностното разпределение на пикселите в определен набор от данни.

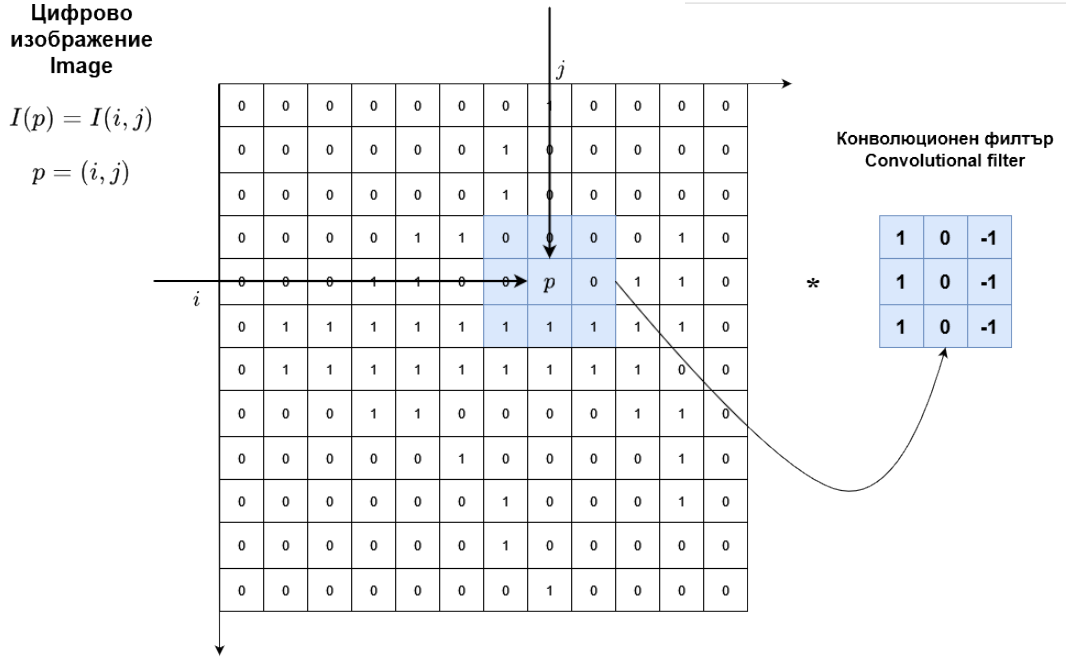
Класическите методи за извличане на контури от структурата на дадено изображение, се извършват посредством числови разлики, изпълнявани по определени схеми, които се наричат диференциални (градиентни оператори).

Проект BG05M2OP001-2.016-0003 „Модернизация на Национален военен университет "Васил Левски"- гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.

Оператори от този вид са метод на Sobel, метод на Roberts, метод на Prewitt, метод на Canny, и метод с оператор на Laplace. Прилагането на различни филтри на едно и също изображение ще произведат различни карти на характеристиките.

След извършване на операцията конволюция се извършва операция по “обединяване”, за да бъде намален броят на параметрите (размерността на невронната мрежа). Същевременно с това се решава и проблема с пренастройване на мрежата. Конволюционните невронни мрежи използват т.нар. „рядка свързаност“, „рядко взаимодействие“ или „редки тегла“ между невроните (sparse connectivity, sparse interactions or sparse weights). Въвеждането на ограничение на броя връзки в мрежата, намалява изискванията на модела относно изчислителна памет, и невроните от крайните слоеве могат индиректно да взаимодействат с тези от входните слоеве. Входните неврони, които индиректно засягат даден изходен неврон, се наричат “рецептивно (възприемчиво) поле” (Receptive Fields) на този изходен неврон. Когато изходният неврон се свързва, с помощта на операция конволюция, с ядро на филтъра с размери ($m \times n$) (ширина и височина), то само $m \times n$ на брой входни неврони засягат конкретният изходен неврон. Обикновено се използват филтри с размери 3 × 3; 5 × 5 или 7 × 7 пиксела, за обхождане на изображението (фиг. 1).

Рецептивното поле се отнася до пространствените измерения на входното изображение и на характеристиките. При подреждане на множество конволюционни слоеве един върху друг, рецептивното поле вече става функция от възприемчивите полета на всички предишни конволюционни слоеве, които се наричат „ефективно рецептивно поле“ (Effective Receptive Fields) за всеки слой.



Фиг. 1. Прилагане на филтърна матрица (3 × 3) към входно изображение I, за откриване на вертикални ръбове и извършване на операция конволюция.

Не на последно място, по значимост, е т.нар. споделяне на параметри (Parameter Sharing, or Shared Weights and Biases). При процеса на обучение на невронната мрежа и изграждане на модела, стойностите на теглата и отклоненията непрекъснато се актуализират с всеки нов пример за обучение. При конволюционните невронни мрежи, стойностите на параметрите се запазват еднакви за всички скрити неврони в даден слой. По този начин всички скрити неврони откриват една и съща характеристика, като ръбове и ъгли, в различни области на изображението. Това прави мрежата толерантна към позицията на обектите в изображението.

В конволюционните мрежи се извършват 3 етапа на обработка на параметрите: конволюционна, нелинейна (добавя се функция на активиране, която да моделира нелинейни функции) и обединяваща, по отношение на прилагане на един филтър, обхождащ изображението:

$$net_{i,j}^{(l-1)} = \sum_{m,n} w_{i,j,m,n}^{(l-1)} x_{i-m,j-n}^{(l-2)} + b_{i,j}^{(l-1)}$$

$$v_{(i,j)}^{(l-1)} = f\left(net_{i,j}^{(l-1)}\right) = \frac{1}{1+\exp^{\left(-net_{i,j}^{l-1}\right)}}$$

$$x_{\frac{i}{2}\frac{j}{2}}^{(l-2)} = \max v_{i-m,j-n}^{(l-1)}, \qquad |m| < \tau, |n| < \tau$$

При прилагане на множество филтри изразите се записват по следния начин:

$$net_{i,j}^{(l-1),k} = \sum_{m,n} w_{i,j,m,n}^{(l-1),k} x_{i-m,j-n}^{(l-2),k} + b_{i,j}^{(l-1),k}$$

$$v_{(i,j)}^{(l-1)k} = f\left(net_{i,j}^{(l-1),k}\right) = \frac{1}{1+\exp^{\left(-net_{i,j}^{l-1,k}\right)}}$$

$$x_{\frac{i}{2}\frac{j}{2}}^{(l-2),k} = \max v_{i-m,j-n}^{(l-1),k}, \qquad |m| < \tau, |n| < \tau$$

където с k се означават броят приложени филтри на слой.

Основните слоеве на конволюционните мрежи са следните:

- слой за въвеждане на изображение (Image Input Layer) – използва се като първи входен слой при невронни мрежи за разпознаване на изображения с един канал за цвят. Входният слой прилага и нормализиране на данните, винаги когато данните се разпространяват напред през него;

- конволюционен слой (Convolution 2D Layer) - задължителен слой за CNN. Различни аргументи като размер на филтри (Filter Size); брой филтри (Number Filters) - брой неврони, които се свързват към същия регион на входа, вкл. и броя на картите на характеристиките; padding за допълване картата на входните характеристики с пиксели, докато се обработва от филтъра или ядрото (Kernel).

- слой за нормализиране на парциални сегменти (Batch Normalization Layer) – нормализира частична извадка на набора от обучаеми образци за всички резултати от наблюдения, независимо за цветовия канал. Нормализира активациите и градиентите, разпространяващи се през невронната мрежа, което оптимизира обучението на мрежата;

- слой коригираща линейна единица (Rectified Linear Units), ReLU layer – съдържа в себе си ReLU функция за активиране, която премахва отрицателните стойности за всеки елемент от входа като изпълнява прагова операция, където всяка стойност по-малка от нула, е зададена на нула;

- обединяващ слой с максимум (Max Pooling 1-D, 2-D Layer) – едномерния обединяващ слой с максимум, приложен върху дадена карта на характеристиките, съкращава размерността и намалява дискретизацията като разделя входа 1-D обединяващи региони, след което операцията изчислява нова матрица от максималните стойности, извлечени от всеки регион;

- напълно свързан слой (Fully Connected Layer) – използва се за решаване на класификационна задача чрез комбиниране на характеристиките за класифициране на изображения. Параметърът „изходен размер“ (Output size) е равен на броя на класовете в целевите данни;

- SoftMax слой (SoftMax Layer) - използва функция за активиране SoftMax, която нормализира изхода получен от напълно свързания слой. Резултатът от SoftMax слоя се състои от положителни числа, които се сумират до единица, в последствие могат да бъдат използвани като класификационни вероятности от класификационния слой;

- класификационен слой (Classification Layer) – изчислява загубата на кръстосана ентропия за класификационни задачи.

При обучаване на невронни мрежи с контролирано обучение, всеки вход има съответен целеви изход, който мрежата се стреми да постигне. За вход подаваме набор от данни, като сравняваме очаквания изход с реално постигнатият резултат от обучение на мрежата. Следваща стъпка е да използваме градиентно спускане за актуализиране параметрите на модела в посока, която ще сведе до минимум разликите между очаквания (оптимален) и действителния резултат, т. нар. обща грешка (total error). Общата грешка се изчилява чрез функция на разходите или целева функция. Тя представлява преобразуване на

една или повече променливи в реално число, като това число представлява „цена“ (cost) или

„загуба“ (loss) на мрежата, наричана още и функция за грешка.

Има много функции на грешките: Средна квадратична грешка (Mean square error - MSE), Квадратична грешка (Squared error - SE), Сума на квадратичните грешки (Sum square error - SSE) и други.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y)^2$$

където

- n – общ брой на обучителните проби;
- $\hat{y}$ – целеви изход;
- y – реален резултат от изходния неврон.

Средноквадратичната грешка се използва за извеждане на изходна величина, с помощта на регресия и с добавяне на функция на загуба, с изчисление на средноквадратична грешка.

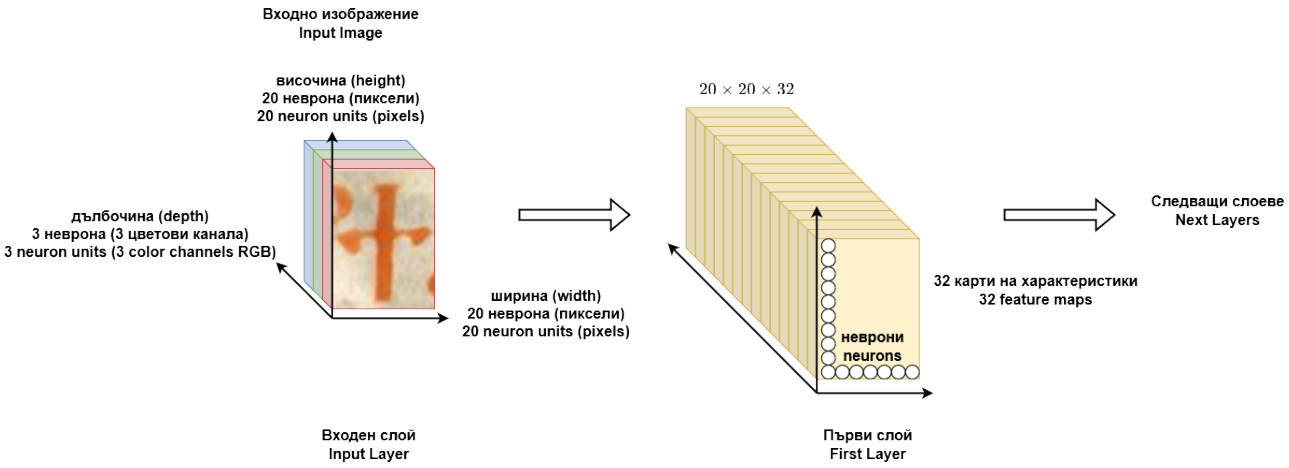
Обучаващите алгоритми са изключително важни параметри за процеса на обучение. В конволюционните невронни мрежи и в Deep Network Designer приложението на MathWorks, за обучение на дълбоки невронни мрежи, са налични за прилагане следните три алгоритъма:

1) Стохастично градиентно спускане с оптимизатор на инерционната константа (SGDM) – итеративен метод за оптимизация на градиентното спускане с оптимизиране на инерционната константа.

2) Корен квадратен от квадратична грешка на разпространение (Root Mean Squared Propagation – RMSProp) - оптимизаторът RMSprop е подобен на алгоритъма за градиентно спускане с инерционна константа. Използва намаляваща средна стойност на частичните градиенти при адаптирането на размера на стъпката за всеки параметър, ограничава колебанията във вертикална посока. Позволява увеличение на скоростта на обучение и алгоритъмът може да предприема по-големи стъпки в хоризонтална посока, сближавайки се по-бързо. Разликата между RMSprop и градиентното спускане, е в начина на изчисление на градиентите. Стойността на импулса по подразбиране се задава със стойност 0.9.

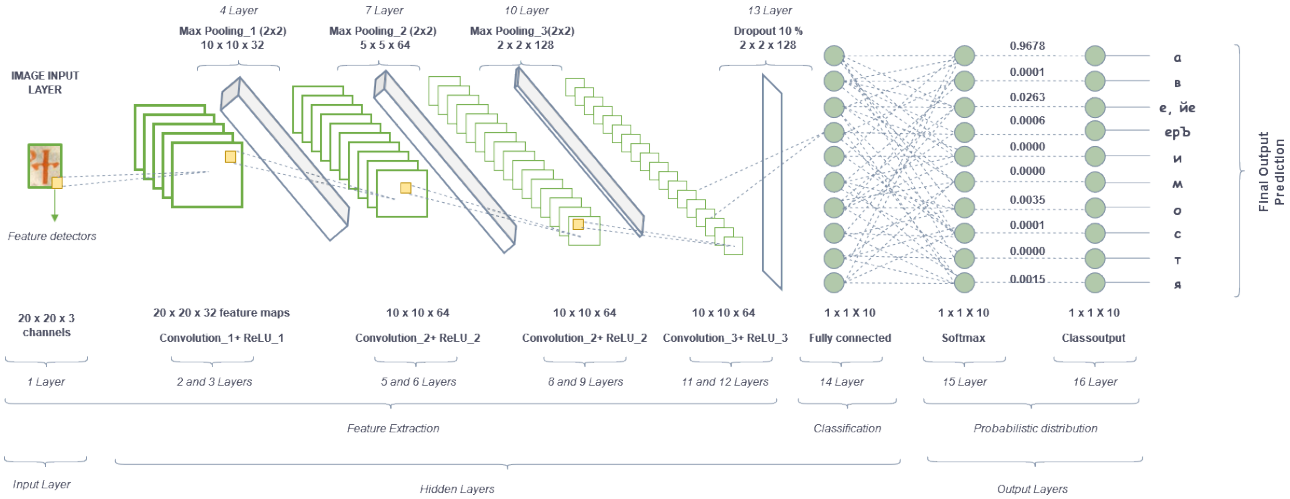
3) Оптимизатор ADAM (Adaptive Moment Estimation – ADAM). Комбиниращ метод и актуализация между оптимизатора RMSProp и градиентно спускане с инерция.

Използването на множество конволюционни слоеве и филтри, в конволюционната невронна мрежа, превръщат изображението в тримерна структура от данни. Това е възможно благодарение на множество свързани карти на характеристики. Тяхното наслагване, една след друга, образува третата ос - по дълбочина d (depth). Филтрите, извличащи характеристиките от изображението, дават информация за интензитета на пикселите в малък регион от изображението, центриран във всеки пиксел, като по този начин се получава информация за всеки пиксел от него. Конволюционните невронни мрежи подреждат своите неврони в три измерения по ширина, височина и дълбочина (width, height, depth – $w \times h \times d$), както е показано на фигура 2. Всеки слой преобразува тримерния входен обем – в тримерен изходен обем. От получените карти на активиране се извличат функции, които намират различни модели в тримерния обем от предходния слой. Непрекъснатото подреждане на тези детектори на характеристики – от по-прости модели на ниско ниво, до сложни обекти от високо ниво, прави т. нар. композиционна йерархия на конволюционните мрежи.

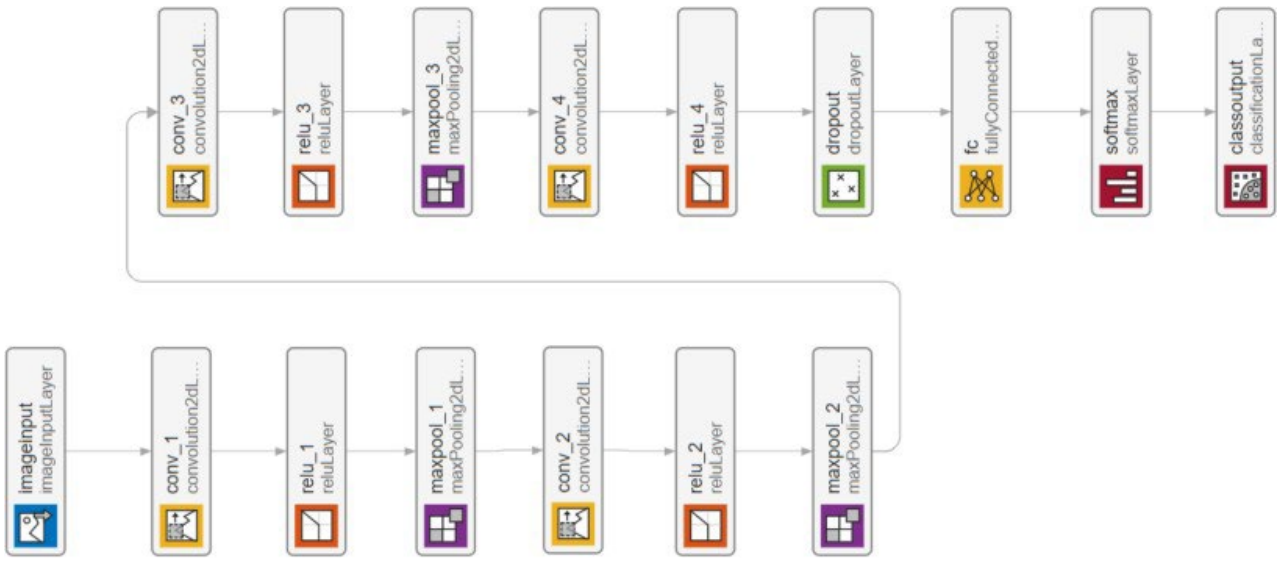


Фиг. 2. Тримерен обем в конволюционните невронни мрежи.

На фиг. 3 е показана архитектура на 16 слойна конволюционна невронна мрежа, обучена да разпознава буквени символи от старобългарската глаголическа азбука от „Зографско четвероевангелие“, базирана на дълбоко структурирано обучение. На фиг. 3а може да се види и конкретната числена стойност, в резултат от разпознаването на буква „а“.



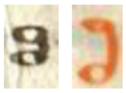
а)



б)

Фиг. 3. Архитектура на конволюционна невронна мрежа за разпознаване на глаголически буквени символи: подробно функционално описание (а) и по слоеве, с приложението Deep Network Designer (б).

Таблица 1: Глаголически букви от Зографското Евангелие – л. 225.

 Буква „а“	 Буква „в“	 Буква „е, йе“	 Буква „и“	 Буква „м“
 Буква „о“	 Буква „с“	 Буква „т“	 Буква „ерЪ“	 Буква „я“

В кодекса е използвано червено мастило за заглавията на евангелията и отделните глави, и черно, и тъмно-кафяво за основния текст Фиг. 4.

а) б)

Фиг.1. Зографско четвероевангелие (източник на изображенията: http://expositions.nlr.ru/ex_manus/Zograph_Gospel/index.php)

Таблица 2: Числени резултати от проведен експеримент по разпознаване на глаголически букви от Асеманиевото и Мариинското евангелия.

Глаголически букви / числен резултат	Асеманиево изборно евангелие – букви кафяв цвят / резултат	Асеманиево изборно евангелие – букви червен цвят / резултат	Мариинско четвероевангелие / резултат
1. „а“	0.9996	1.0000	0.9362
2. „в“	0.9727	0.9686	неразпознат
3. „е, йе“	0.9713	неразпознат	0.9997
4. „и“	неразпознат	неразпознат	неразпознат
5. „м“	0.9468	неразпознат	неразпознат
6. „о“	0.8584	0.9736	неразпознат
7. „с“	0.8266	неразпознат	0.9914
8. „т“	0.3738	0.6888	0.9992
9. „ерЪ“	0.9498	0.9922	неразпознат
10. „я“	0.9998	1.0000	1.0000

Числените резултати посочени в таблица 2 са за буквен символ, който е разпознат и има най-висока стойност сред вероятностните резултати за останалите букви.