



## РАЗПОЗНАВАНЕ НА ИЗОБРАЖЕНИЯ С ИЗПОЛЗВАНЕ НА КОНВОЛЮЦИОННА НЕВРОННА МРЕЖА

Задачата, свързана с разпознаване на обекти в изображения, може да се сведе до задача за сравнение между наблюдавано изображение и еталонно изображение (входове от тестов набор от данни). В съответствие с определени критерии се извършва класификация на обектите, която може да отнесе обекта към определен клас.

За обучението на изкуствените невронни мрежи и тестване на производителността на моделите се използват три типа набори от данни:

- набор от данни за обучение (training dataset) – използва се за коригиране на стойностите на тегловите матрици;
- набор от данни за валидиране (validation dataset) – използва се за минимизиране на проблем, известен като прекомерно напасване (настройване) към данните (overfitting) и помага за избора на обучителни алгоритми и параметри;
- набор от данни за тестване (testing dataset) – с тестовия набор се проверява колко точно е обучена изкуствената невронна мрежа.

Задачата за разпознаване на обекти в изображения може да се отнесе към следните задачи от полето на машинното обучение - класификация и класификация с липсващи входни величини.

При задачата за контролирано обучение като класификация, обученият алгоритъм посочва на изходният слой на невронната мрежа към коя от категориите  $k$  принадлежи дадена входна величина.

Алгоритъмът на обучение се представя с набор от входни величини, заедно с желаните от тях резултати (наричани още етикети – labels), като изпълнява следната функция:

$$f: \mathbb{R}^n \rightarrow \{1, \dots, k\},$$

където  $\mathbb{R}$  е множество на всички реалните числа за  $k = 1, 2, \dots, n$ ; с  $k$  се обозначава клас или категория, към която принадлежи дадена входна величина.



При  $y = f(x)$ , алгоритъмът за обучение присвоява входни величини, описани с вектор  $x$ , в категория  $y$ .

При решаване на задачата за класификация с липсващи входове, или т. нар. неконтролирано обучение, алгоритъмът се представя с набор от входове, но без да са посочени желани изходи, което означава, че алгоритъмът трябва да намира самостоятелно структури и модели от множество. Алгоритъмът за обучение трябва да дефинира еднофункционално съпоставяне на входен вектор към изходна категория и вместо да предоставя една функция за класификация следва да научи набор от функции, като всяка функция съответства на класифициране на  $x$  с липсващо подмножество, различно от нейните входове.

Наборът от данни (dataset) е колекция от много примери, с която се учи алгоритъма за обучение. Цифровите изображения са от съществено значение за машинното обучение и изкуствените невронни архитектури.

Освен големите набори от данни, необходими за обучението на DNNs, е важна и сложността на самото изображение, т.е. да се позволи на изкуствената невронна мрежа да се учи от големия брой функции, с които са дефинирани пробите, екземплярите (samples, instances)) в рамките на набора от данни.

Обучението „без учител“, включва наблюдение на няколко примера за произволен вектор  $x$  и опит за изрично научаване на вероятностното разпределение  $p(x)$ .

Обучение „с учител“ включва наблюдение на няколко примера за случаен вектор  $x$  и свързана стойност или вектор  $y$  и се научава да прогнозира  $y$  от  $x$ , обикновено чрез оценка на  $p(y|x)$ .

За решаване на поставената задача, е необходимо разработването на големи набори от данни, с които оптимално да се обучат моделираните сложни невронни архитектури.

Приема се, че обектите от еталонните модели на самолети и тези от възстановени радиолокационни изображения са инвариантни по отношение на мащаб и ориентация в кадъра. Създадените еталонни модели (цифрови изображения) са подложени на последваща цифрова обработка чрез внасяне на различни нива на добавен шум в изображенията, което симулира претрупана околна среда с електромагнитни смущения.

Предполага се, че нивото на шумове варира по случаен закон и в широки граници. Ако наблюдението на обекта продължава с последователно

[www.eufunds.bg](http://www.eufunds.bg)

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.



генериране на множество радарни изображения, то изображенията на обекта се различават значително едно от друго, заради неговото движение. С наслагване на различни по вид и съотношение шум върху еталонните модели се генерират множество изображения на всеки от двадесет и двата летателни обекта. Всяко изображение симулира регистрирани различни на брой елементарни отражатели по повърхността на обекта, намиращи се в различно кинематично състояние, описвайки отделни части (точки) от неговата пространствена геометрия.

За генериране на наборите от данни е използван итеративен подход.

Използваните смущения са адитивен Гаусов шум с нормално разпределение (white Gaussian noise) и импулсен шум (Salt & pepper noise). Всеки еталонен модел на самолет (двоична матрица на изображение  $A$ ) се зашумява с внасяне на адитивен шум, като на всяка итерация се променят характеристиките на шума със стъпка на промяна 0.01.

Наборите от данни се описват като множество съдържащо  $m$  елементи:

$$\{I^{(1)}, I^{(2)}, \dots, I^{(m)}\},$$

където  $I$  е едно изображение от хомогенното множество, тъй като размерността на всички изображения е еднаква.

Създадени са три бази от данни с изображения с добавен гаусов шум и три с импулсен шум.

При генерирането на всеки набор с изображения се изменят характеристиките на съответния шум последователно както е посочено: от 0 до 0.51 със стъпка на промяна 0.01; от 0 до 0.71 със стъпка на промяна 0.01 и от 0 до 0.91 със стъпка на промяна 0.01.

Създадените набори от данни и техните характеристики са представени в Таблица 1, в зависимост от характеристиките на използваните два вида шум. С използването на различен брой екземпляри се цели постигането на приблизително еднаква големина на наборите от данни.



Таблица 1. Описание на създадените набори от изображения.

Име на набора от данни	Параметри на използвания шум	Нарастване на параметрите в диапазона	Брой екземпляри за всяка итерация	Брой изображения за всеки модел самолет	Общ брой изображения на набора от данни
1.database_with_Gaussian_noise_v0.52_m0.52_3	математическо очакване (mean [m]) и дисперсия (variance [v])	мат. очакване [m] и дисперсия [v] от 0 до 0.51	3	8 427	185 394
2.database_with_Gaussian_noise_v0.72_m0.72_2	математическо очакване (mean [m]) и дисперсия (variance [v])	мат. очакване [m] и дисперсия [v] от 0 до 0.71	3	10 368	228 096
3.database_with_Gaussian_noise_v0.92_m0.92_1	математическо очакване (mean [m]) и дисперсия (variance [v])	мат. очакване [m] и дисперсия [v] от 0 до 0.91	1	8 464	186 208
1.database_with_Salt&Pepper_noise_d0.52_160	плътност на разпределението (noise density [d])	плътност на разпределение [d] от 0 до 0.51	160	8 480	186 560
2.database_with_Salt&Pepper_noise_d0.72_120	плътност на разпределението (noise density [d])	плътност на разпределение [d] от 0 до 0.71	120	8 760	192 720
3.database_with_Salt&Pepper_noise_d0.92_90	плътност на разпределението (noise density [d])	плътност на разпределение [d] от 0 до 0.91	90	8 370	184 140

Броят изображения за всеки модел е постигнат от комбинирането на параметрите на съответния шум. Всеки екземпляр е създаден след като към определения еталонен модел е разпределен шум на случаен принцип, но с еднакви стойности на параметрите.

Присвояването на всеки пиксел на произволна вероятностна стойност от разпределението на шума в един модел не съвпада с разпределението на шума в останалите модели. Приблизителното изравняване в разпределението на

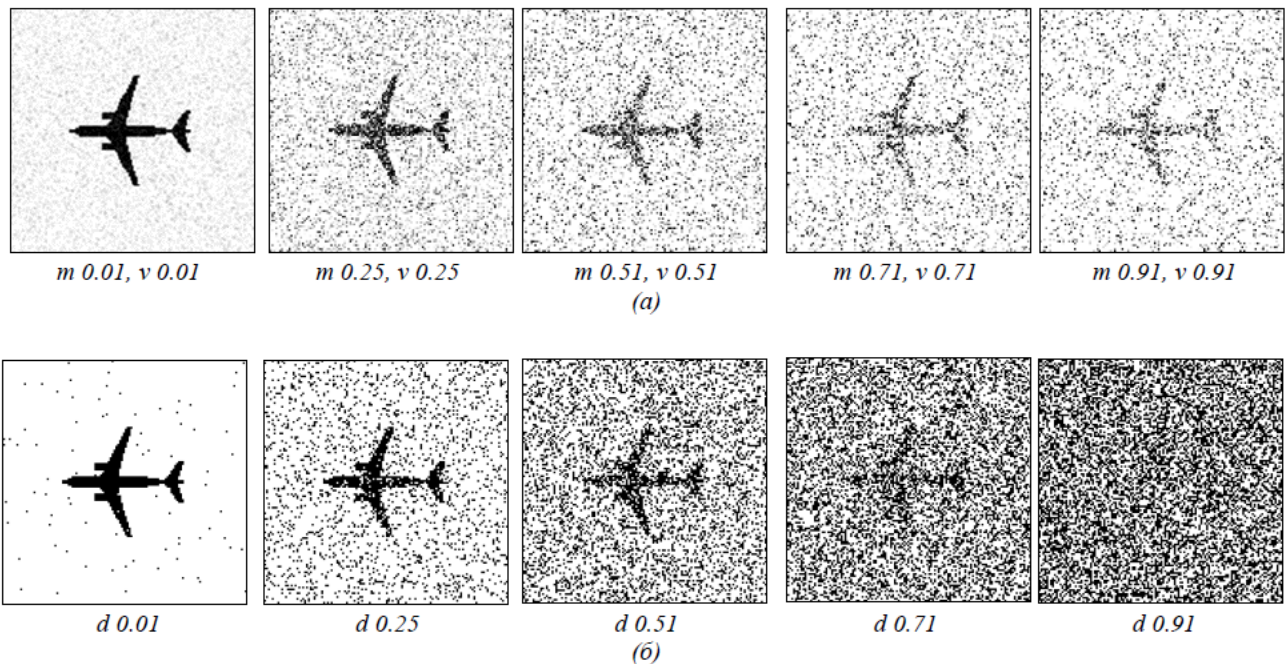
[www.eufunds.bg](http://www.eufunds.bg)

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.



данните за обучение в различните бази от данни, както и в броя на изображенията по класове е за да бъде избегнато пристрастие към определени свръх представени класове.

На фигура 1 са посочени примерни изображения за модел An-158 от създадените бази с изображения с адитивен гаусов шум (фиг. 1а) и импулсен шум (фиг. 2б) с различна степен на постъпково зашумяване.



Фиг. 1. Създадени изображения на самолет An-158 в присъствие на различни нива адитивен гаусов шум (а) и импулсен шум (б)

Конволюционните невронни архитектури за разпознаване на изображения са предназначени за работа с двумерни изображения от едномерни или тримерни данни. Те извличат и научават отделни характеристики от изображението слой по слой, съхраняват информацията в отделни неврони, след което обединяват научените характеристики за да разпознаят цялостното изображение.

Изборът и изграждане на мрежови архитектури ще има следната последователност от действия – дефиниране на архитектурата на модела, определяне на параметрите на модела. Мрежовите архитектури варират в зависимост от броя на включените слоеве и използваните параметри. Изборът на архитектури е аналогичен при изграждане на невронни архитектури и за двата





вида добавен шум. Слоевете, които са използвани за обучение на реализираните модели архитектури, са стандартни за конволюционните мрежи.

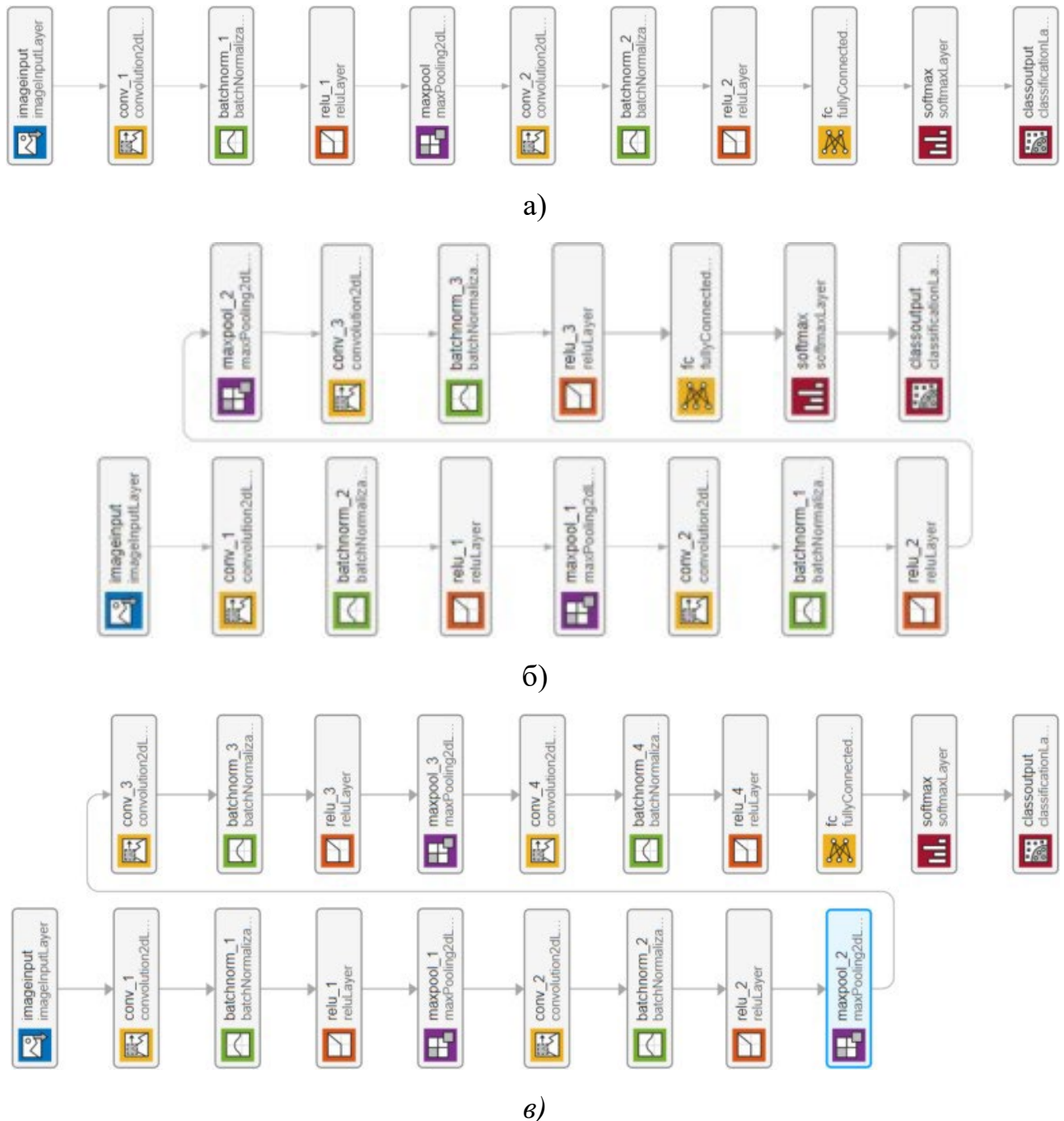
Първоначално е избрана архитектура с 11 слоя, тъй като класовете за обучение са малко на брой и изображенията не са хроматични.

Използваните слоеве за обучение на моделираните конволюционни мрежи са следните:

- слой за въвеждане на изображение (Image Input Layer) – входен слой 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) – с посочени параметри като размер на филтри (Filter Size), брой филтри (Number Filters), padding за допълване картата на входните характеристики с пиксели;
- 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;
- слой коригираща линейна единица (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;
- едномерен обединяващ слой с максимум (Max Pooling 1-D Layer) – едномерния обединяващ слой с максимум извършва намаляване на дискретизацията като разделя входа на едноразмерни обединяващи региони, след което изчислява средната стойност за всеки регион;
- напълно свързан слой (Fully Connected Layer) – използва се за решаване на класификационна задача като комбинира характеристиките за класифициране на изображенията. 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.

На фигура 2 са изобразени архитектурите на конволюционна невронна мрежа с използване на различен брой слоеве.



Фиг. 2. Мрежова архитектура на конволюционна невронна мрежа:  
11 слойна (а), 15 слойна (б) и 19 слойна (в).



В двата конволюционни слоя са използвани 32 броя филтри с размери  $3 \times 3$ , със стъпка на изместване (Stride) [1 1], с който се движи филтъра, прилагайки вертикално и хоризонтално отместване с по един пиксел.

Обединяващият слой намалява броя на стойностите в картите на характеристиките. Избрана е размерността да бъде намалена чрез максимално обединяване, използвайки максимална стойност на площта (Pooling size) [2 2] и размер на стъпката също [2 2].

Друг подход за намаляване на размерността чрез обединяващ слой е използването на обединяване чрез средна стойност, което не е приложено в настоящата работа. На фигура 3 може подробно да се разгледа описаната по-горе невронната архитектура за отделните слоеве и направени настройки.

ANALYSIS RESULT				
	Name	Type	Activations	Learnables
1	imageinput 128×128×1 images with 'zerocenter' normalization	Image Input	128×128×1	-
2	conv_1 32 3×3×1 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	128×128×32	Weights 3×3×1×32 Bias 1×1×32
3	batchnorm_1 Batch normalization with 32 channels	Batch Normalization	128×128×32	Offset 1×1×32 Scale 1×1×32
4	relu_1 ReLU	ReLU	128×128×32	-
5	maxpool 2×2 max pooling with stride [2 2] and padding 'same'	Max Pooling	64×64×32	-
6	conv_2 32 3×3×32 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	64×64×32	Weights 3×3×32×32 Bias 1×1×32
7	batchnorm_2 Batch normalization with 32 channels	Batch Normalization	64×64×32	Offset 1×1×32 Scale 1×1×32
8	relu_2 ReLU	ReLU	64×64×32	-
9	fc 22 fully connected layer	Fully Connected	1×1×22	Weights 22×131072 Bias 22×1
10	softmax softmax	Softmax	1×1×22	-
11	classoutput crossentropyex with 'a Alenia M-346' and 21 other classes	Classification Output	1×1×22	-

Фиг. 3. Мрежова архитектура и настройка на параметри на отделните слоеве на 11-слойната невронна мрежа.





Като част от експерименталните резултати ще бъдат изследвани разликите в производителността на обучените конволюционни невронни мрежи с еднаква архитектура и различни обучителни алгоритми.

Използваните обучителни алгоритми 'sgdm', 'adam' и 'rmsprop' са вградени за приложението Deep Network Designer.

Друг хиперпараметър, който ще бъде изследван е скоростта на обучение. Разширените обучителни параметри на скоростта на обучение като намаляване на скоростта на обучение по време на обучението (Learning Rate Schedule), брой епохи за отпадане на скоростта на обучение (Learning Rate Dropout Period), коефициент за намаляване на скоростта (Learning Rate Dropout Factor), коефициент за регулиране на L2, са зададени със стойности по подразбиране и няма да бъдат предмет на експеримент.

Инерционната константа е със стойност 0.9 и се прилага за всички реализирани модели невронни мрежи в настоящия труд.

Максималният брой епохи за обучение са зададени на 30 с използване на мини-партиди от по 128 резултата за наблюдение на всяка итерация.

Средата за изпълнение е избрана да бъде с използване на графичен процесор.

**В процеса на обучение** на конволюционни невронни мрежи, изображенията се филтрират в конволюционни слоеве чрез конволюционни филтри, създавайки карти на активации (карта на характеристиките). Автоматичното научаване на голям брой паралелни филтри в конволюционните мрежи, характерни за конкретна задача и набор от данни, ги прави водещи при невронните архитектури за решаване на специфични проблеми с прогнозно моделиране като класификация на изображения.

Резултатите за набора от данни database\_with\_Gaussian\_noise\_v0.52\_m0.52\_3 с използването на различни стойности за скорост на обучението и различни обучителни алгоритми е представено в Таблица 2.

На таблицата са представени и резултатите от шест обучени невронни мрежи с посочени време за обучение и точност на валидиране.

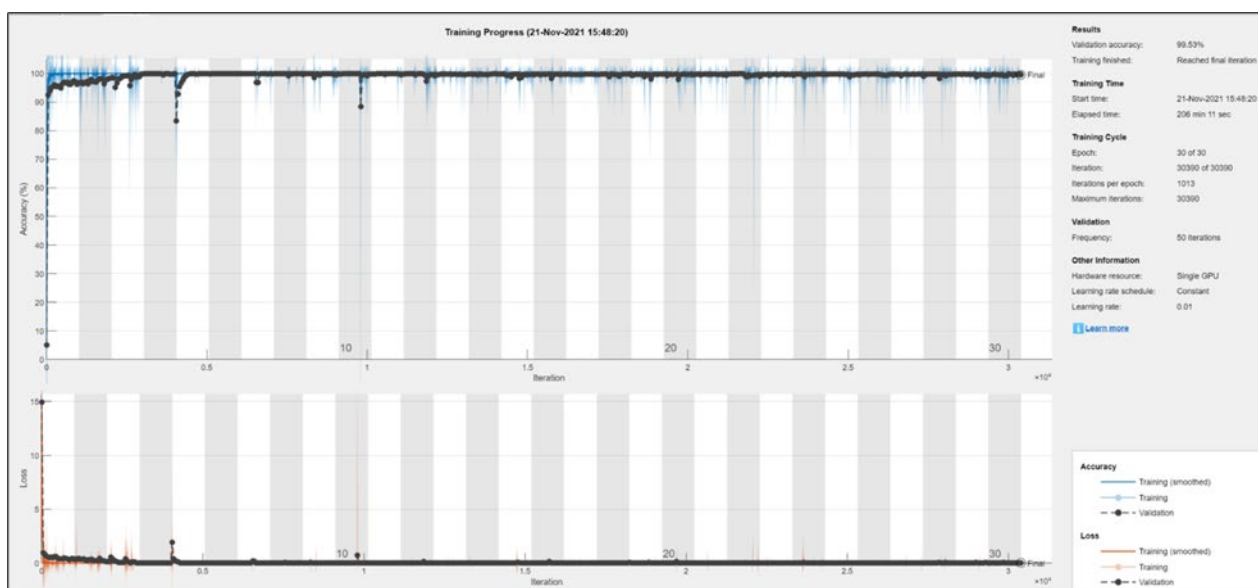


Таблица 2. Обучение на CNNs с различни обучителни алгоритми и скорост на обучение.

CNN	Обучаващ алгоритъм SGDM	Обучаващ алгоритъм ADAM	Обучаващ алгоритъм RMSPROP
	Точност на валидиране [%] / Време за обучение [min, s]		
11 слоя скорост на обучение 0.01	99.24 % 213 min 45 sec	99.53 % 206 min 11 s	99.48 % 203 min 2 s
11 слоя скорост на обучение 0.001	100 % 214 min 5 sec	99.98 % 198 min 11 sec	99.99 % 219 min 49 s

Всички 6 на брой обучени мрежи, посочени в таблицата, са достигнали 100 % точност на обучение (Training accuracy) и са намалили загубите на трениране и валидиране до минимум (Training Loss, Validation Loss).

Напредъкът на обучението може да се наблюдава в използваното приложение за дълбоко обучение на Matlab с помощта на динамично начертване на различни показатели по време на процеса, както е показано на фигура 3.



Фиг. 3. Обучителен прогрес на 11-слойна CNN с трениращ алгоритъм ADAM и скорост на обучение 0.01.



Точността на обучение и валидиране на описаните в таблицата модели е в синхрон, което показва, че моделите са добре обучени и не се е стигнало до прекомерно напасване на модела (overfitting).

По правило точността на валидиране следва винаги да е малко по-ниска от тази на обучението. Това може да бъде обяснено с факта, че данните за валидиране са колекция от изображения, които са нови за обучаващия се модел. Не винаги най-добрият модел, който постига оптимални резултати при валидиране ще е и най-добър при тестване. Тази разлика може да бъде свързана с това, че тестовите данни идват от друг независим източник или може да бъде свързана с размера и качеството на тестовия набор от данни.

Същото е валидно да се твърди и за валидиращите данни. Данни, които не са на нивото на сложност на обучителните и са подадени за валидиращи, могат да покажат висок резултат в края на обучението и да дадат погрешна представа за цялостното представяне на модела. Въпреки това, процесът на валидиране може да бъде приет като още една проверка на производителността на модела и да бъде насочваща. Ако точността на обучение, на валидиране и на тестване съвпадат, тогава е постигнато оптимално обучение на модела.

Следващите експерименти продължават с обучение на нови невронни мрежи, но с различен брой слоеве и еднакви параметри на обучение. Наборът от данни, който е използван за обучението им е database\_with\_Gaussian\_noise\_v0.52\_m0.52\_3. В таблица 3 е посочена мрежовата структура на конволюционните невронни мрежи.

*Таблица 3. Подробна структура на отделните слоеве и приложени параметри на обучение на CNNs.*

Номер на слоя	11-слойна CNN	15-слойна CNN	19-слойна CNN
1	Image Input [128 128 1]	Image Input [128 128 1]	Image Input [128 128 1]
2	Conv_1 [128 128 32]; 3x3x32 Filter size [3 3]; Filter number 32 stride [1 1]; padding [1 1 1]	Conv_1 [128 128 32]; 3x3x32 Filter size [3 3]; Filter number 32 stride [1 1]; padding [1 1 1]	Conv_1 [128 128 32]; 3x3x32 Filter size [3 3]; Filter number 32 stride [1 1]; padding [1 1 1]
3	Batchnorm_1 [128 128 32]; 32 channels	Batchnorm_1 [128 128 32]; 32 channels	Batchnorm_1 [128 128 32]; 32 channels
4	ReLU_1 [128 128 32]	ReLU_1 [128 128 32]	ReLU_1 [128 128 32]
5	Maxpool_1 [64 64 32] [2 2] max pool, stride [2 2], padding 'same'	Maxpool_1 [64 64 32] [2 2] max pool, stride [2 2], padding 'same'	Maxpool_1 [64 64 32] [2 2] max pool, stride [2 2], padding 'same'
6	Conv_2 [64 64 32]; 3x3x32 Filter size [3 3]; Filter number 32	Conv_2 [64 64 32]; 3x3x32 Filter size [3 3]; Filter number 32 stride [1 1]; padding [1 1 1]	Conv_2 [64 64 32]; 3x3x32 Filter size [3 3]; Filter number 32 stride [1 1]; padding [1 1 1]

[www.eufunds.bg](http://www.eufunds.bg)

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.



	stride [1 1]; padding [1 1 1 1]		
7	Batchnorm_2 [64 64 32] 32 channels	Batchnorm_2 [64 64 32] 32 channels	Batchnorm_2 [64 64 32] 32 channels
8	ReLU_2 [64 64 32]	ReLU_2 [64 64 32]	ReLU_2 [64 64 32]
9	FC 22 [1 1 22]	Maxpool_2 [32 32 32] [2 2] max pool, stride [2 2], padding 'same'	Maxpool_2 [32 32 32] [2 2] max pool, stride [2 2], padding 'same'
10	Softmax [1 1 22]	Conv_3 [32 32 32]; 3x3x32 Filter size [3 3]; Filter number 32 stride [1 1]; padding [1 1 1 1]	Conv_3 [32 32 32]; 3x3x32 Filter size [3 3]; Filter number 32 stride [1 1]; padding [1 1 1 1]
11	Classoutput [1 1 22]	Batchnorm_3 [32 32 32] 32 channels	Batchnorm_3 [32 32 32] 32 channels
12		ReLU_3 [32 32 32]	ReLU_3 [32 32 32]
13		FC 22 [1 1 22]	Maxpool_3 [16 16 32] [2 2] max pool, stride [2 2], padding 'same'
14		Softmax [1 1 22]	Conv_4 [16 16 32]; 3x3x32 Filter size [3 3]; Filter number 32 stride [1 1]; padding [1 1 1 1]
15		Classoutput [1 1 22]	Batchnorm_4 [16 16 32] 32 channels
16			ReLU_4 [32 32 32]
17			FC 22 [1 1 22]
18			Softmax [1 1 22]
19			Classoutput [1 1 22]

На таблица 4 е показан резултата от обучението на CNNs с различен брой слоеве. Двете невронни мрежи с петнадесет и деветнадесетслойна архитектура също са обучени с алгоритъм на обучение SGDM и скорост на обучение 0.001. Всички мрежи са достигнали 100 % точност на обучение и са намалили загубите на трениране и валидиране до минимум в края на обучителния период.

Таблица 4. Обучение на CNNs с различен брой слоеве и еднакви параметри на обучение.

CNNs Брой слоеве	Точност на валидиране [%]	Време за трениране [min, s]
11-слойна	100 %	214 min 5 sec
15-слойна	99.98 %	204 min 23 sec
19-слойна	99.94 %	197 min 22 sec



Последният етап се състои в обучаването на други две CNNs с 11 слойна архитектура. За обучаващи данни са използвани трите вида набори от данни с зашумени изображения на гаусов шум. Резултати са показани в Таблица 3.5.

*Таблица 5. Обучение на CNNs с различен вид набори от данни и еднаква архитектура и параметри на обучение.*

Набор от данни	11 слойна CNNs	
	Точност на валидиране [%]	Време за трениране [min, s]
<i>database_with_Gaussian_noise_v0.52_m0.52_3</i>	100 %	214 min 5 sec
<i>database_with_Gaussian_noise_v0.72_m0.72_2</i>	99.99 %	312 min 19 sec
<i>database_with_Gaussian_noise_v0.92_m0.92_1</i>	99.86 %	208 min 10 sec

От проведените експерименти може да се направи извод, че прилагането на различни видове обучителни алгоритми и различна скорост на обучение е правилен подход за търсене на оптимален модел на невронна архитектура. Както може да се види от резултатите, броят на слоевете не указва почти никакво влияние върху точността на валидиране.

Времето за обучение варира, но е в близки граници. При набора от данни с *database\_with\_Gaussian\_noise\_v0.72\_m0.72\_2* процесът е по-дълъг, което се дължи на разликата от близо **42 000 изображения в повече**, от останалите два набора от данни за този шум.

Сложността в изображенията на различните набори от данни също не указва затруднение при обучението на моделите. Крайният резултат от работата по разпознаване на обекти от изображения и провеждането на числени експерименти с обучените модели, ще покаже влиянието на различните обучителни алгоритми, хиперпараметри, брой слоеве и набори от данни. Моделираните конволюционни мрежи в настоящата работа имат от един до три конволюционни слоя в архитектурата си. Хиперпараметърът, който асоциира изходното изображение към входния регион и произвежда характеристиката е рецептивното поле (receptive field) – размера на филтъра. Всички конволюционни слоеве са със зададени 32 броя филтри.

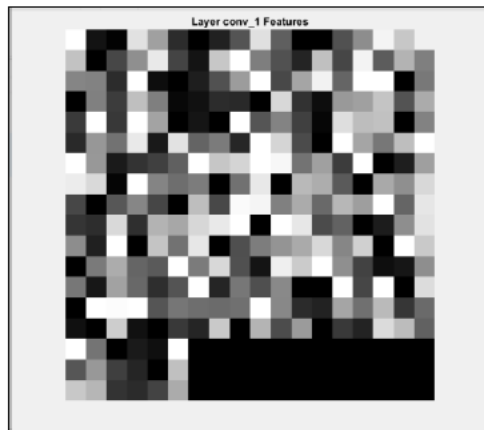
Първият конволюционен слой научава характеристиките на ниско ниво, а с увеличаване на дълбочината на мрежата, слоевете към края ѝ научават



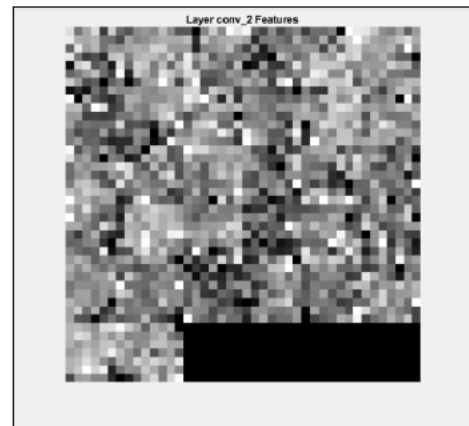


характеристики от по- високо ниво. Това е ключов фактор за разбирането на принципа на научаване на вътрешните слоеве на тези модели архитектури.

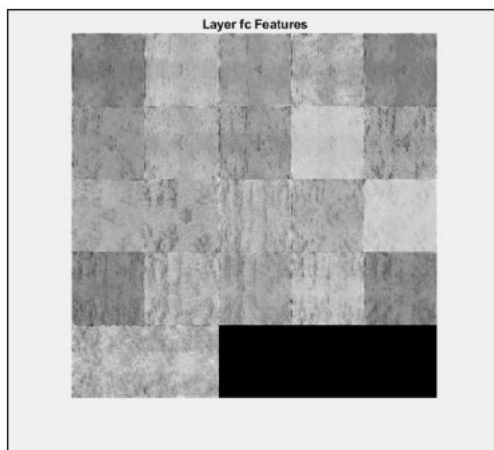
С помощта на функции в Deep Learning Toolbox на MathWorks и техниката за визуализация Deep Dream Image, на фиг. 4 са представени визуализации на картите на характеристики от различни слоеве на конволюционна мрежа, обучена за разпознаване на изображения с внесен гаусов шум.



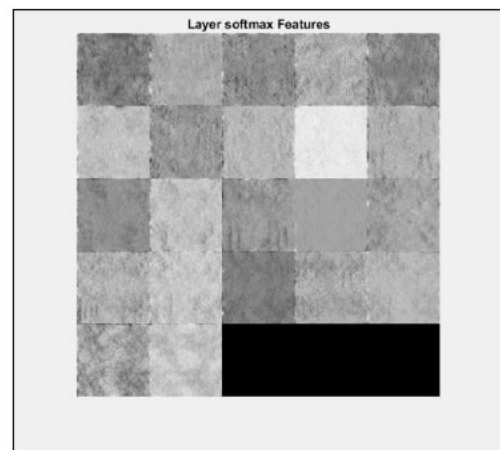
(a)



(б)



(в)



(г)



д)

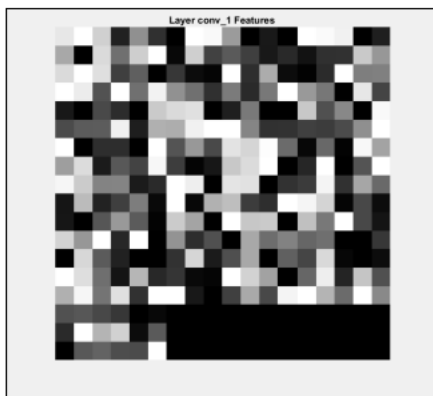
*Фиг. 4. Визуализиране на картите на характеристики на 11-слойна CNN, обучена за разпознаване на адитивен гаусов шум по слоеве: конвол. слой 1 (а), конвол. слой 2 (б), напълно свързан слой (в) и SoftMax слой (г) за двадесет и двата класа самолета, и отново напълно свързан слой с изведени 6 класа самолети (д).*

В единадесетслойната конволюционна мрежа има два конволюционни слоя – 2-ри (фиг. 4а) и 6-ти слой (фиг.4б), които научават характеристиките на различни нива, в конкретния случай - втори слой на ниско и шести слой на високо ниво. Рецептивните полета в обучените модели конволюционни невронни мрежи за всички конволюционни слоеве е 32 броя. Изображенията, с които са визуализирани конволюционните слоеве съдържат предимно нюанси на цвят в сивата скала и структури в изображенията. Това показва, че в конкретната задача, за която са обучени мрежите това са филтри предимно за нюансите в сивата скала и детектори на структури, които извличат картите на характеристиките от изображенията на летателни обекти с внесен шум. Визуализирането на напълно свързания слой (фиг. 4в) създава изображения, които най-много наподобяват двадесет и двата класа самолети, и които силно активират избраните класове.

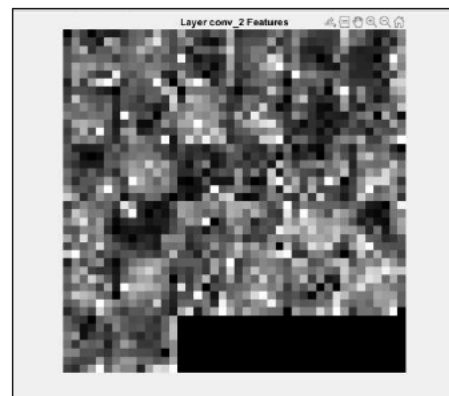


Аналогично могат да се визуализират картите на характеристиките на деветнадесет слойна мрежа за разпознаване на изображения с внесен импулсен шум. В структурата на деветнадесетслойната конволюционна мрежа има 4 конволюционни слоя и един напълно свързан слой. Повечето слоеве ще представят и по подробни филтри, които са научили сложните модели и текстури от изображенията от набора от данни. Следва също да се вземе предвид, че използваната обучена мрежа за визуализиране на картите на характеристиките на фигура 4 е обучена с набор от данни с най-висок импулсен шум - database\_with\_Salt&Pepper\_noise\_d0.92\_90.

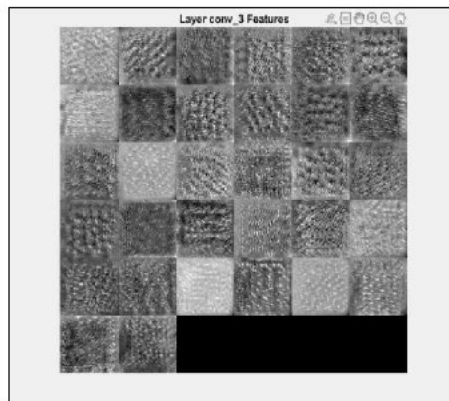
Възможността за обучение на конволюционна мрежа с повече слоеве и по-голяма дълбочина, помага на невронната мрежа да изучи повече представяния от обучителния набор. За да може да се обучи мрежата да разпознава изображения в условия на много високи нива на шум, е необходимо обучителните набори от данни да се състоят точно от такива екземпляри. 11-слойната невронна мрежа (фиг. 4.) освен, че се обучава да класифицира летателни обекти, анализира и шума в изображенията и това е видимо на фигура 4в - в напълно свързания слой, в картите на характеристиките освен структурата на шума (представена във вид на текстура) могат да се доловят и очертанията на летателните обекти по класове. Докато 19-слойната конволюционна мрежа на фиг. 5 почти напълно неутрализира добавения импулсен шум.



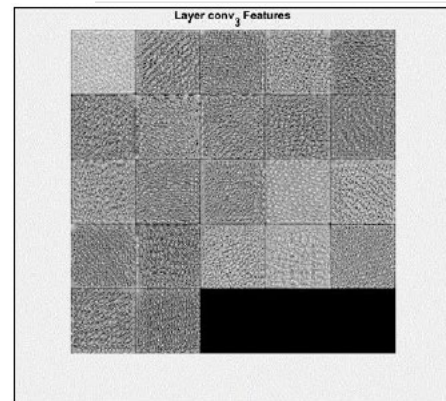
(a)



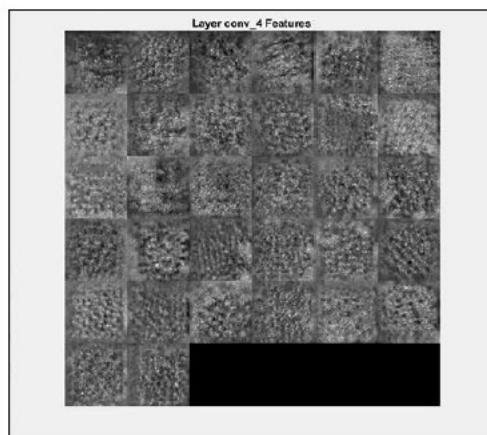
(б)



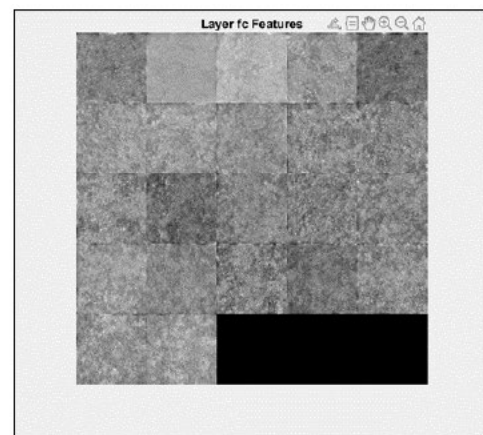
(в)



(г)



(д)



(е)

Фиг. 5. Визуализиране на картите на характеристики на 19-слойна CNN, обучена за адитивен импулсен шум по слоеве: конвол. слой 1 (а), конвол. слой 2 (б), конвол. слой 3 (в), конвол. слой 3 по класове (г), конвол. слой 4 (д) и напълно свързан слой по класове(е) за двадесет и двата класа самолета.

Конволюционните невронни мрежи могат да покажат и активациите на различните слоеве на мрежата, сравнявайки областите на активиране с оригиналното изображение, както е показано на фиг. 6 и фиг. 7.



ЕВРОПЕЙСКИ СЪЮЗ  
ЕВРОПЕЙСКИ  
СОЦИАЛЕН ФОНД



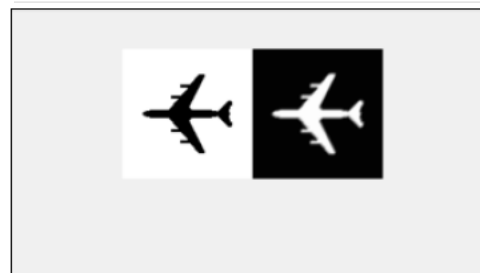
ОПЕРАТИВНА ПРОГРАМА  
НАУКА И ОБРАЗОВАНИЕ ЗА  
ИНТЕЛИГЕНТЕН РАСТЕЖ



(a)

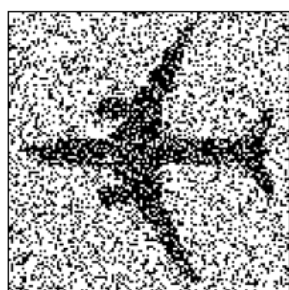


(б)

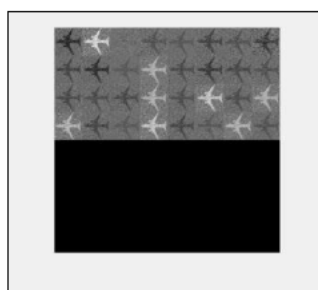


(в)

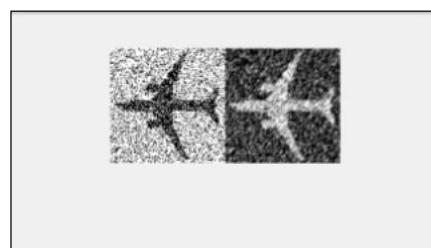
Фиг. 6. Представяне на функциите на активации в първи конволуционен слой на изображение на Boeing 707 без добавен шум (a), активации на всеки от 32-та филтъра (б), активации в 32-я канал на слоя.



(a)



(б)



(в)

Фиг. 7. Представяне на функциите на активации в първи конволуционен слой на изображение на Boeing 787-8 с импулсен шум с плътност на шума 0.5 (a), активации на 32-та филтъра (б), активации в 32-я канал на слоя.

Подадените изображения на невронната мрежа връщат изходните активации на първия конволуционен слой, както е показано на по-горните фигури. Активациите се връщат като тримерен масив, като третото измерение индексира канала на конволиционния слой. За да бъдат показани тези активации следва третото измерение да се преоразмери в четиримерен масив. Третото измерение представлява цвета и тъй като изображенията са едноканални и активациите нямат цвят, стойността на третото и четвъртото измерение имат стойности единица, като четвъртото измерение индексира канала.

Всички активации са мащабираны така, че минималното активиране да е 0, а максималното да е 1.

[www.eufunds.bg](http://www.eufunds.bg)

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.