

*rozpoznawanie koloru, neuronowy detektor,
implementacja sieci neuronowej, automatyka przemysłowa*

Dariusz WAWRZY尼亚K, Marcin KAMIŃSKI*

SPRZĘTOWA IMPLEMENTACJA NEURONOWEGO DETEKTORA KOLORÓW

W artykule przedstawiono detektor długości fal świetlnych, wykorzystujący teorię sztucznych sieci neuronowych. Przedstawiono opis struktury oraz projektowania detektora, a także aplikację sprzętową analizowanego modelu neuronowego. W części praktycznej niniejszego projektu zastosowano mikrokontroler ATmega32. Wykonano badania prezentujące dokładność działania detektora neuronowego przy współpracy z czujnikiem koloru.

1. WPROWADZENIE

Ciągły rozwój automatyzacji procesów przemysłowych wymusza, aby maszyny i urządzenia posiadały zdolności do szybkiego rozpoznawania barw i ich odcieni. Kolor jest postrzegany przez ludzkie oko poprzez falę elektromagnetyczną. Jednym z najistotniejszych problemów projektowych w systemach automatyki było, do niedawna, rozpoznawanie kształtów i kolorów badanych obiektów. Aktualnie spotykane rozwiązania czujników koloru wykorzystują jedną z dwóch metod pomiaru barwy. Pierwsza polega na aktywnym oświetleniu badanej powierzchni oraz pomiarze natężenia odbitych promieni przy użyciu prostego przetwornika. Aktywne oświetlenie polega na naprzemiennym oświetlaniu obiektu światłem czerwonym, zielonym i niebieskim [1]. Druga metoda pomiaru polega na wykorzystaniu detektora składającego się z matrycy przetworników posiadających filtry czerwone, zielone i niebieskie. Badana powierzchnia oświetlana jest światłem białym. Czujniki wykorzystujące tę metodę nazywane są czujnikami *true color* [2].

Proste sensory nie umożliwiają bezpośredniego odczytu koloru badanej powierzchni a jedynie porównanie barwy z wzorcowymi wartościami składowych RGB, wprowadzonych do pamięci czujnika. Bardziej rozbudowane czujniki wykorzystują

* Politechnika Wroclawska, Instytut Maszyn, Napędów i Pomiarów Elektrycznych, ul. Smoluchowskiego 19, 50-372 Wrocław, e-mail: 191564@student.pwr.wroc.pl, marcin.kaminski@pwr.edu.pl

metody analitycznego wyznaczania współrzędnych x , y przestrzeni barw. Takie rozwiązanie pozwala na odczytanie koloru z wykresu chromatyczności [3], [4]. Jednak prezentowany w publikacjach diagram stanowi formalny opis reprezentacji barw, nieuwzględniający wielu czynników rzeczywistych (np. związanych z warunkami pomiaru lub rzeczywistą implementacją). W praktycznych aplikacjach korzystne może być zastosowanie sieci neuronowej. Sposób projektowania modelu neuronowego nie wymaga dokładnego poznania wzorca, modelu matematycznego oraz parametrów obiektu [5]–[8]. W zastosowaniach przemysłowych istotną cechą jest wrażliwość czujników na wszelkie zmiany położenia i własności badanej powierzchni, dlatego muszą być kalibrowane do wykonywania pomiarów w powtarzalnych warunkach. W celu poprawienia precyzji rozpoznawania kolorów, w przemyśle, stosuje się rozbudowane układy optyczne oraz dodatkowe układy kompensujące wpływ zakłóceń zewnętrznych na wyniki pomiarów. W amatorskich rozwiązaniach stosowanie takich układów jest bardzo ograniczone ze względów praktycznych oraz ekonomicznych.

W niniejszym artykule opisany został czujnik koloru typu *true color* współpracujący z neuronowym detektorem długości fali światła. Sensor wykorzystujący matrycę detekcyjną TCS3200 zawiera oświetlacz złożony z czterech białych diod LED. Celem przeprowadzonych testów było przedstawienie możliwości zastosowania SSN w zagadnieniach dotyczących detekcji koloru. Sztuczna sieć neuronowa została wytrenowana *off-line* w środowisku Matlab, następnie algorytm został zaimplementowany (w języku C/C++) w mikrokontrolerze Atmega32. Moduł rozpoznający kolory, wykorzystany został do budowy urządzenia do sortowania elementów kolorowych.

2. PROJEKT MODELU NEURONOWEGO

Neuronowy detektor długości fali światła został oparty o dwuwarstwową, jednokierunkową sieć neuronową. Wartościami wejściowymi SSN są wartości częstotliwości zmierzone na czterech kanałach czujnika koloru. Pierwsze trzy wartości wejściowe reprezentują zawartość składowych czerwonej, zielonej oraz niebieskiej w strumieniu światła odbitego od badanej powierzchni. Czwarta zmienna określa wartość natężenia oświetlenia. Cztery wartości wejściowe przetwarzane są przez dwie warstwy ukryte, z których pierwsza składa się z pięciu, a druga z ośmiu neuronów. Obie warstwy ukryte zbudowane są z neuronów typu sigmoidalnego, w których funkcją aktywacji jest tangens hiperboliczny. W warstwie wyjściowej znajduje się jeden neuron o liniowej funkcji aktywacji.

Uczenie SSN przeprowadzone zostało w środowisku Matlab przy użyciu pakietu Neural Network. Jako zbiór uczący wykorzystano serię rzeczywistych wartości zmierzonych czujnikiem koloru. Pomiarów dokonano na wzorniku składającym się z jedenastu kolorowych pól o określonych wartościach długości fal. Dla każdego pola wykonane zostały trzy pomiary wartości składowych mierzonych przez czujnik. Pomiary

wykonywane były w obudowie chroniącej układ pomiarowy przed wpływem zewnętrznego oświetlenia. Częstotliwości prostokątnych sygnałów wyjściowych detektora zawierały się w przedziale od 10 do 15000 Hz. W celu uzyskania najlepszej efektywności działania nieliniowych funkcji aktywacji, wartości wejściowe normalizowano, zostały podzielone przez 15000. Aby wielkościami wyjściowymi SSN były liczby znormalizowane, wartości długości fal zostały podane w mikrometrach. Tak przygotowany zbiór uczący został przedstawiony sztucznej sieci neuronowej 200 razy. Do wytrenowania sieci neuronowej wykorzystany został algorytm Levenberga–Marquardta.

Algorytm Levenberga–Marquardta należy do najskuteczniejszych metod optymalizacji parametrów sieci neuronowych. W wymienionej metodzie uczenia wykorzystuje się aproksymowaną wartość hesjanu, będącego macierzą drugich pochodnych funkcji celu względem współczynników wagowych. Aproksymacji hesjanu dokonuje się na podstawie informacji zawartych w gradiencie, przy uwzględnieniu czynnika regularyzującego.

Minimalizowana w trakcie treningu funkcja opisana jest poniższym równaniem [9]:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N [e_i(\mathbf{w})]^2 \quad (1)$$

oraz

$$e_i = [y_i(W) - d_i], \quad (2)$$

gdzie N – oznacza liczbę wyjść. Przy wprowadzeniu zapisu macierzowego:

$$\mathbf{e}(\mathbf{w}) = \begin{bmatrix} e_1(\mathbf{w}) \\ e_2(\mathbf{w}) \\ \dots \\ e_N(\mathbf{w}) \end{bmatrix}, \quad (3)$$

$$\mathbf{J}(\mathbf{w}) = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \dots & \frac{\partial e_1}{\partial w_n} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \dots & \frac{\partial e_2}{\partial w_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_N}{\partial w_1} & \frac{\partial e_N}{\partial w_2} & \dots & \frac{\partial e_N}{\partial w_n} \end{bmatrix}, \quad (4)$$

aproksymowana macierz hesjanu:

$$\mathbf{H} = \mathbf{J}^T(\mathbf{w})\mathbf{J}(\mathbf{w}) \quad (5)$$

Adaptacja wag sieci neuronowej jest realizowana iteracyjnie, zgodnie z zależnością:

$$\Delta \mathbf{w} = -(\mathbf{H}(\mathbf{w}))^{-1} \mathbf{J}^T(\mathbf{w}) \mathbf{e}(\mathbf{w}) \quad (6)$$

W trakcie realizacji obliczeń może wystąpić problem z odwrotnością macierzy hesjanu, w celu rozwiązania tego problemu wprowadza się poniższą definicję tej macierzy [10]:

$$\mathbf{H} = \mathbf{J}^T(\mathbf{w}) \mathbf{J}(\mathbf{w}) + \mu \mathbf{I} \quad (7)$$

zatem równanie opisujące wprowadzane poprawki wag sieci neuronowej:

$$\Delta \mathbf{w} = -(\mathbf{J}(\mathbf{w})^T \mathbf{J}(\mathbf{w}) + \mu \mathbf{I})^{-1} \mathbf{J}^T(\mathbf{w}) \mathbf{e}(\mathbf{w}) \quad (8)$$

Obliczenia macierzy \mathbf{J} realizowane są według algorytmu wstecznej propagacji błędu. Parametr μ jest współczynnikiem adaptacji decydującym o oddziaływaniu algorytmu adaptacyjnego.

3. IMPLEMENTACJA SZTUCZNEJ SIECI NEURONOWEJ W MIKROKONTROLERZE ATMEGA32

Implementacja sztucznej sieci neuronowej w mikrokontrolerze polega na stworzeniu funkcji symulującej przetwarzanie modelu neuronowego oraz wprowadzeniu do pamięci wartości poszczególnych wag oraz elementów *bias*. Wszystkie współczynniki wagowe wewnętrznych warstw sieci zostały zgrupowane w dwóch macierzach. Natomiast wartość *bias* wraz z wagami połączeń warstwy wyjściowej zapisane zostały w oddzielnym wektorze. Obliczenia przedstawione w poprzedniej części artykułu zostały zrealizowane w środowisku Matlab, uzyskane wartości modelu, otrzymane w wyniku działania algorytmu uczonego, zostały zapisane do pamięci mikrokontrolera. Na potrzeby implementacji SSN, w pamięci sterownika, utworzone zostały trzy wektory przeznaczone do przechowywania wartości wejściowych dwóch warstw ukrytych oraz wartości wejściowych warstwy wyjściowej. Pierwsze komórki tych wektorów reprezentują wejścia polaryzujące neuronów, a ich wartość jest zawsze równa jedności. Zapis do wektorów rozpoczynany jest od drugiej komórki. Dodatkowo utworzone zostały dwa wektory pomocnicze, wykorzystywane do przechowywania wartości wyjściowych bloków sumacyjnych poszczególnych neuronów, warstw ukrytych. Pozwala to na rozdzielaniu operacji wykonywanych przez bloki sumacyjne oraz bloki aktywacji poszczególnych neuronów. Realizacja algorytmu SSN, na dowolnej platformie zapewniającej przetwarzanie szeregowe, polega na przemnażaniu wartości wejściowych neuronów przez wartości wag, sumowaniu wszystkich wartości wejściowych tych neuronów oraz obliczeniu, na ich podstawie, wartości wyjściowych

ich funkcji aktywacji. Uzyskanie przetwarzania równoległego w realizacji praktycznej jest możliwe przy wykorzystaniu matryc FPGA [11].

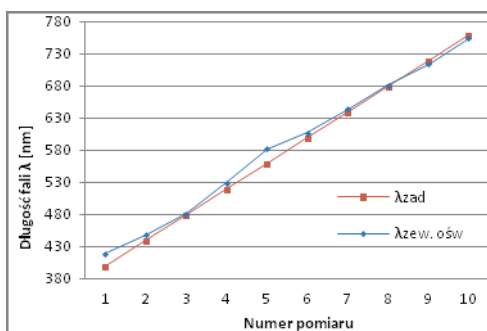
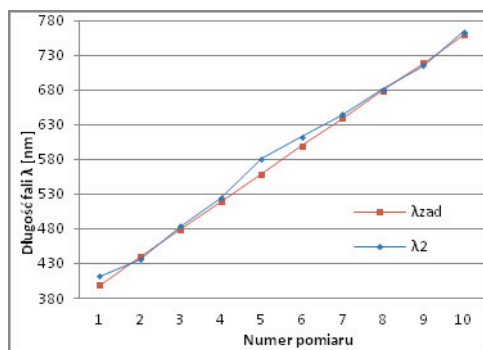
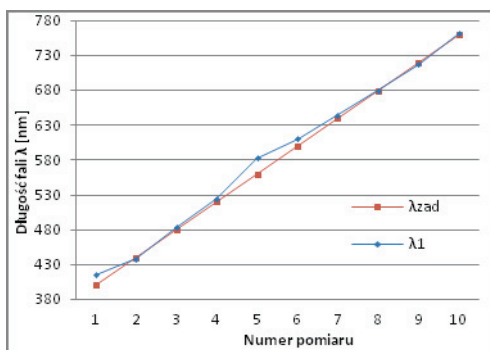
Z pośród dostępnych metod przeprowadzenia pomiarów wartości częstotliwości przy użyciu mikrokontrolera, wybrana została metoda zliczania impulsów [12]. Podczas jednego pomiaru długości fali światła, wartości częstotliwości sygnałów zmierzonych dla czterech kanałów czujnika koloru dzielone są przez 15000, po czym zapisywane zostają do wektora wejściowego pierwszej warstwy ukrytej. Następnie wektor ten jest przemnażany przez wartości wag i współczynników *bias* pierwszej warstwy ukrytej. Otrzymane wartości są sumowane, po czym zapisywane do wektora pomocniczego o wymiarze zgodnym z liczbą neuronów tej warstwy. Kolejna operacja polega na obliczeniu wartości wyjściowych bloków aktywacji. Argumenty funkcji pochodzą z wektora pomocniczego. Wyniki tej operacji zapisywane są w wektorze wejściowym kolejnej warstwy. Te same operacje wykonywane są dla drugiej warstwy ukrytej. Po obliczeniu wartości wejściowych warstwy wyjściowej obliczana jest suma iloczynu tego wektora z wektorem wag i poziomu *bias* warstwy. Wynikiem tej operacji jest długość fali światła padającego na matrycę detekcyjną czujnika koloru. Wartość ta jest następnie mnożona przez 1000, aby otrzymać wynik pomiaru podany w nanometrach.

4. BADANIA DETEKTORA

W następnym etapie prac testom poddano neuronowy algorytm wyznaczający długość fali na podstawie pomiaru częstotliwości. Informacja o każdym z analizowanych kolorów zawarta jest w długości fali, z zakresu 400–760 nm. Na podstawie tablicy, w której kolory posortowano według długości fal światła, odczytano kod określający położenie kolorów w przestrzeni RGB. W programie graficznym wpisując kody uzyskano kolory, z których następnie wykonano wzorniki wykorzystywane w generowaniu danych uczących oraz testujących. Zbiór danych testujących, nie był uwzględniony w procesie treningu sieci neuronowej. Dla kolejnych rzeczywistych wartości długości fali λ_{zad} odczytywano wskazania sieci neuronowej. Wykonano dwa pomiary długości fali λ_1 oraz λ_2 w warunkach zbliżonych do pomiaru składowych podczas budowy zbioru uczącego. W ten sposób zaprezentowano powtarzalność uzyskiwanych wyników. Następnie wykonano pomiar długości fali $\lambda_{\text{zew.ośw.}}$ w warunkach, w których układ pomiarowy narażony był na wpływ zewnętrznego oświetlenia. Wyniki pomiarów długości fal światła, odbitych od powierzchni wzornika, przedstawione zostały w tabeli 1, graficzna prezentacja wyników została zamieszczona na rysunku 1. Przeprowadzone testy potwierdzają skuteczność działania zaprojektowanego algorytmu neuronowego współpracującego z przetwornikiem światło–częstotliwość.

Tabela 1. Wyniki pomiaru długości fal światła odbitych od wzornika

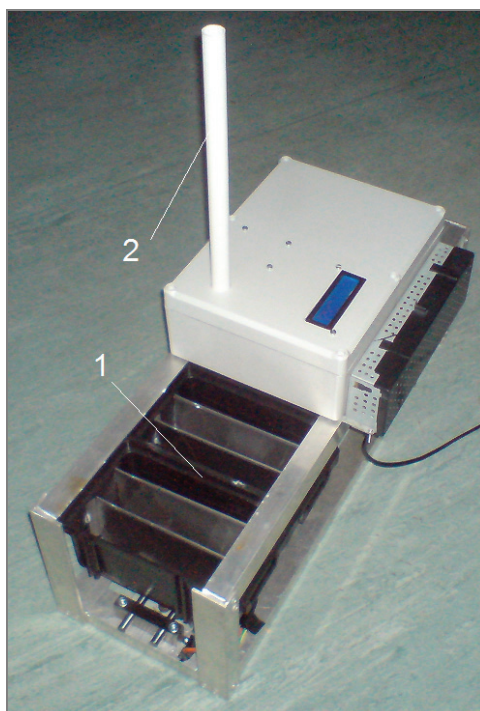
Lp.	λ_{zad}	λ_1	λ_2	$\lambda_{\text{zew. ośw.}}$
	[nm]	[nm]	[nm]	[nm]
1	400	415	413	420
2	440	438	437	449
3	480	483	485	482
4	520	525	525	530
5	560	583	582	583
6	600	611	613	608
7	640	645	646	645
8	680	680	682	682
9	720	718	716	715
10	760	762	765	755



Rys. 1. Porównanie wartości rzeczywistych długości fali oraz wskazywanych przez algorytm neuronowy

5. SORTOWNIK ELEMENTÓW KOLOROWYCH

Opisany w poprzedniej części artykułu algorytm neuronowy współpracujący z czujnikiem koloru TCS3200 zastosowano w rzeczywistym modelu urządzenia klasyfikującego elementy kolorowe. Ogólną budowę zaprezentowano na rysunku 2. Jednym z głównych elementów konstrukcji jest pojemnik, w którym umieszczone są posortowane elementy (podzielony na sześć równych komór). W poszczególnych komorach znajdują się elementy posegregowane według kolorów. Na rysunku 2 widoczny jest również zasobnik, w którym przechowywane są detale przeznaczone do sortowania.

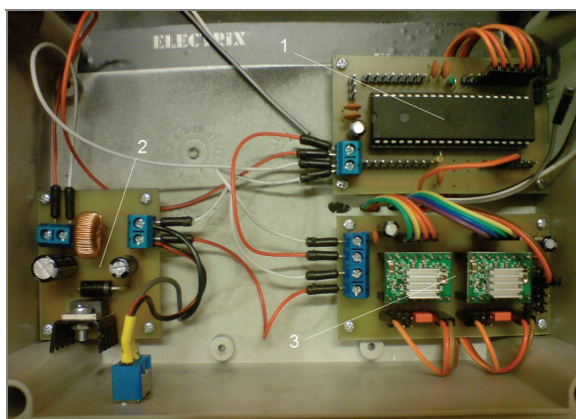


Rys. 2. Urządzenie do sortowania elementów kolorowych
(1 – pojemnik z przegrodami, 2 – zasobnik)

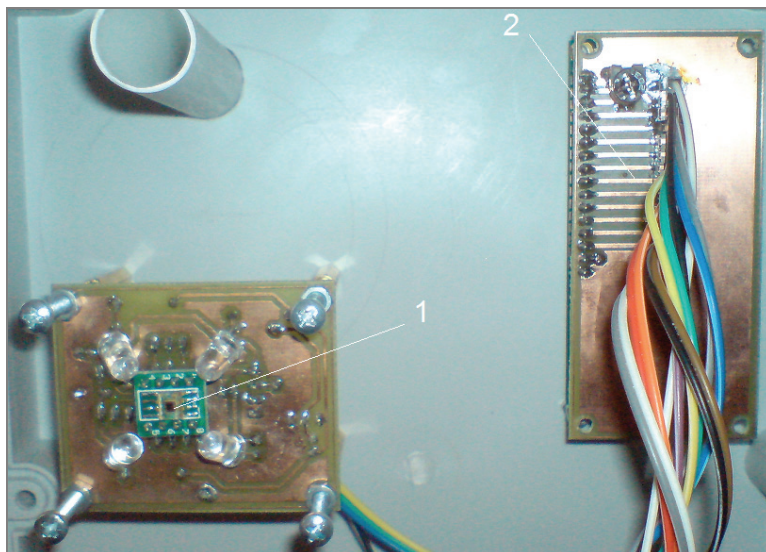
Projekt zakładał wykonanie w pełni automatycznego urządzenia do sortowania elementów kolorowych. W układzie detekcji kolorów zastosowano model neuronowy oraz czujnik TCS3200. Czujnik zawiera oświetlacz o regulowanym natężeniu, emitujący białe światło oświetlające badany przedmiot. Jako układ przetwarzająco-sterujący wykorzystano mikrokontroler ATmega32. Na podstawie sygnałów pochodzących z czujnika,

sieć neuronowa zaimplementowana w mikrokontrolerze, dokonuje klasyfikacji koloru. Rozpoznanie barwy polega na estymacji długości fali światła odbitego od badanego elementu. Założono, że elementami segregowanymi są kolorowe kulki o średnicy 12 mm. Pojedynczy element posiada jednolitą barwę, a wszystkie kulki wykonane są z tego samego materiału oraz charakteryzowały się taką samą strukturą powierzchni. Na podstawie danych uzyskanych na wyjściu sieci neuronowej dokonywane jest sortowanie elementów przy użyciu odpowiednio sterowanych układów napędowych. W części wykonawczej użyto dwóch silników krokowych sterowanych przez mikrokontroler za pośrednictwem scalonych sterowników, zawierających podwójne mostki H. W układzie zasilania części wykonawczej zastosowano zasilacz o napięciu pracy dostosowanym do napięcia zasilającego urządzenia wykonawcze. Część logiczna urządzenia zasilana była za pośrednictwem dodatkowego stopnia obniżającego napięcie do standardu TTL. Urządzenie zapewnia możliwość obserwacji przebiegu procesu sortowania za pośrednictwem wyświetlacza LCD 16x2. Projekt zakładał wykonanie oddzielnych obwodów elektronicznych, z których każdy znajdował by się na oddzielnej płycie drukowanej.

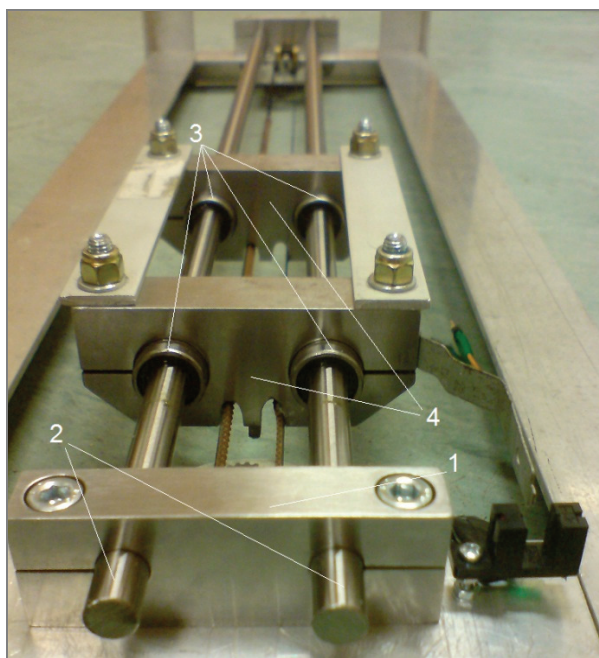
Po uruchomieniu urządzenie automatycznie ustala położenie układów wykonawczych. Proces sortowania rozpoczynany jest od pobrania kolorowego elementu z zasobnika i umieszczenia go w polu widzenia czujnika. Następnie, układ detektora koloru dokonuje pomiaru trzech składowych RGB. Dodatkowo wykonywany jest pomiar natężenia oświetlenia. Na podstawie zebranych danych, sieć neuronowa rozpoznaje długość fali światła padającego na matrycę detekcyjną czujnika. Po zaklasyfikowaniu elementu do jednej z sześciu grup, urządzenie odpowiednio ustawia pojemnik przechowujący posortowane elementy (prowadnica liniowej napędzana przekładnią z pasem zębatym). Następnie, aktualnie badany element umieszczany jest w odpowiedniej komorze pojemnika. Po zakończeniu tej czynności, proces rozpoczynany jest od początku. Wybrane szczegóły konstrukcyjne zaprezentowano na rysunkach 3–8.



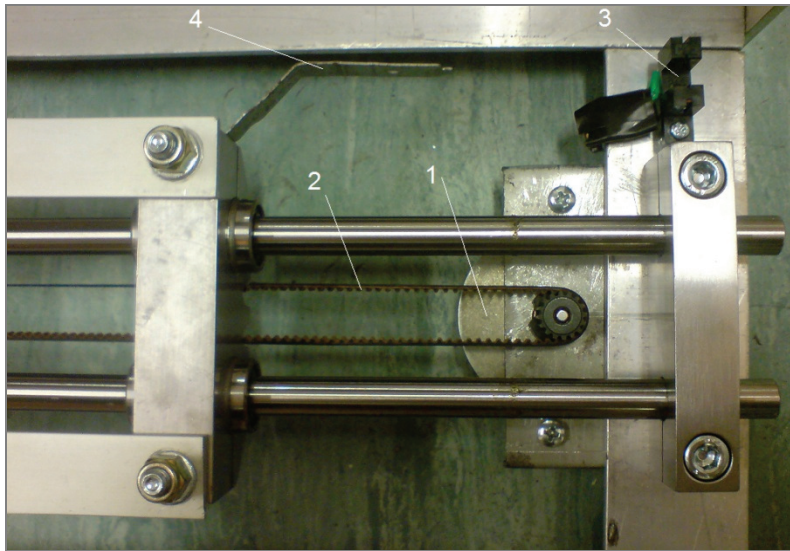
Rys. 3. Obwody elektroniczne zamontowane w obudowie podajnika
(1 – mikrokontroler, 2 – przetwornica impulsowa, 3 – płyta sterowników silników krokowych)



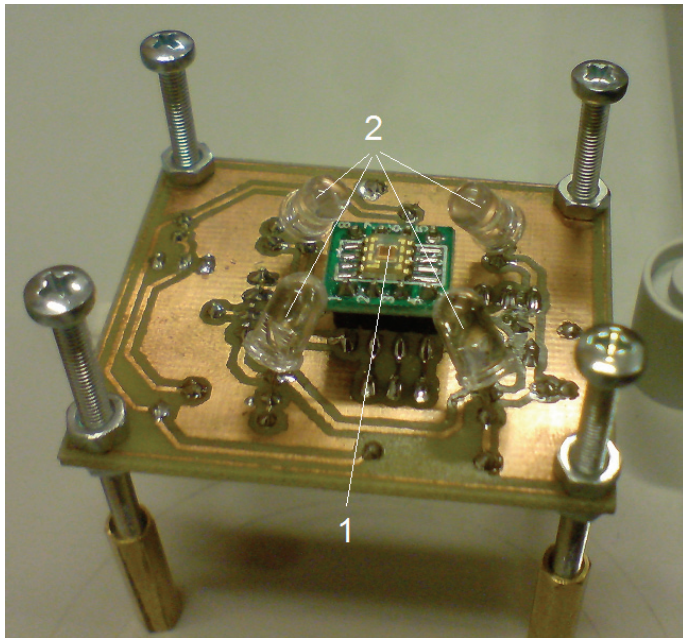
Rys. 4. Obwody elektroniczne zamontowane w pokrywie obudowy podajnika
(1 – czujnik koloru, 2 – płytki pomocnicza wyświetlacza LCD)



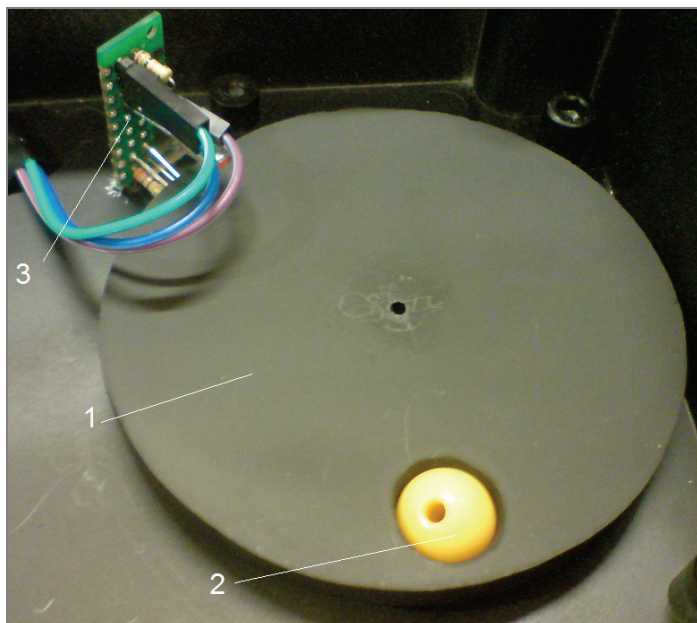
Rys. 5. Kompletny zespół napędu liniowego (1 – podpora prowadnic, 2 – pręty precyzyjne, 3 – łożyska liniowe toczne, 4 – obudowy łożysk liniowych)



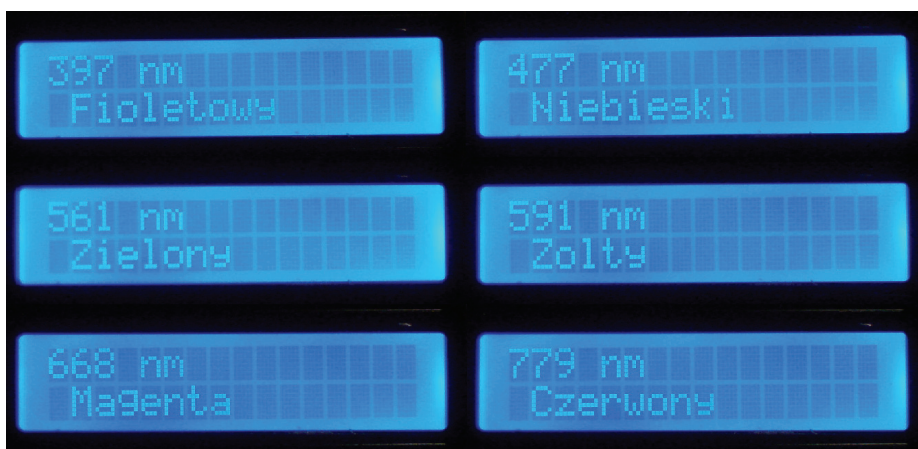
Rys. 6. Układ napędowy pojemnika (1 – silnik krokowy, 2 – przekładnia z pasem zębatym, 3 – optyczny czujnik krańcowy, 4 – znacznik dla czujnika krańcowego)



Rys. 7. Czujnik koloru zamontowany w pokrywie obudowy podajnika (1 – czujnik TCS3200, 2 – układ oświetlający, diody LED)



Rys. 8. Podajnik wraz z elementem przeznaczonym do sortowania (1 – obrotowa tarcza, 2 – testowany element, 3 – czujnik optyczny służący do pozycjonowania podajnika)



Rys. 9. Wyniki przykładowych pomiarów długości fal światła odbitych z rzeczywistych elementów wraz z nazwami przedziałów (do których zostały zaklasyfikowane)

Przykładowe wartości długości fal wyznaczone przy użyciu detektora neuronowego przedstawiono na rysunku 9. Dodatkowo, na wyświetlaczu LCD wyświetlone zostały nazwy przedziałów, do których zostały zaklasyfikowane sortowane elementy.

Dla badanego zakresu kolorów oraz przedstawionych próbek, we wszystkich przypadkach, detale zostały prawidłowo zaklasyfikowane.

6. PODSUMOWANIE

W artykule opisano zagadnienia związane z wykorzystaniem sztucznej sieci neuronowej do detekcji długości fali światła na podstawie wartości składowych RGB mierzonych czujnikiem koloru. Ponadto dla zaprezentowanej struktury detektora neuronowego przeprowadzone zostały badania sprawdzające skuteczność jego działania. Przedstawione wyniki prezentują precyzję działania detektora. Zaproponowana metodyka implementacji, wykorzystująca sieć neuronową, jest wygodna do praktycznej implementacji. Poprzez zastosowanie sieci neuronowej możliwe jest, bezpośrednio na podstawie pomiaru częstotliwości składowych RGB, wyznaczanie barwy elementu. Jest to znaczne uproszczenie algorytmu szczególnie dla większej ilości rozpoznawanych kolorów. Szczególnie korzystne wydaje się zastosowanie detektora neuronowego w warunkach niepewności pomiaru oraz zakłóceń zewnętrznych. Opisany projekt modelu neuronowego może być traktowany jako przykład według którego możliwe jest opracowanie detektora dla większej liczby barw. Ostatecznie neuronowy detektor koloru zastosowano w modelu sortownika elementów kolorowych, wykonano model którego silniki krokowe sterowane są na podstawie informacji o wykrytej długości fali. Całościowo urządzenie może być rozbudowywane według potrzeb aplikacyjnych. W literaturze branżowej o zasięgu międzynarodowym, można spotkać zastosowania czujnika TCS3200 współpracującego z aplikacją uruchamianą na komputerze PC [13]. Zaprezentowano również detektor neuronowy, składający się z dwóch modeli [14]. Pierwszy z nich zastosowano w celu wygenerowania przesłanek o kolorach (sieć Kohonena), drugi wykrywał kolor (sieć wielowarstwowa). Na podstawie powyższych informacji można stwierdzić oryginalność zaproponowanego rozwiązania.

LITERATURA

- [1] PASZKIEWICZ T., *E3MC RGB Color Sensor Przemysłowe czujniki koloru*, Elektronika Praktyczna, 2001, Vol. 2, 132–135.
- [2] HENDRYCH B., *Rozpoznawanie barw. Aplikacje czujników koloru MEE*, Elektronika Praktyczna, 2012, Vol. 1, 112–114.
- [3] KUMAR A., CHOUDHURY R., *Principles of Colour and Appearance Measurement: Object Appearance, Colour Perception and Instrumental Measurement*, Elsevier, 2014.
- [4] VELHO L., FRERY A.C., GOMES J., *Image Processing for Computer Graphics and Vision*, Springer Science & Business Media, 2009.
- [5] HUANG W., QIAO Y., TANG X., *Robust Scene Text Detection with Convolution Neural Network Induced MSER Trees*, Computer Vision – ECCV 2014, 13th European Conference, 2014, Vol. 8692, 497–511.

- [6] MARKOU M., SINGH S., *Novelty detection: a review – Part 2: neural network based approaches*, Signal Processing, 2003, Vol. 83, No. 12, 2499–2521.
- [7] ROWLEY H.A., BALUJA S., KANADE T., *Neural network-based face detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, Vol. 20, No. 1, 23–38.
- [8] TADEUSIEWICZ R., *Sieci neuronowe*, Akademicka Oficyna Wydawnicza, Warszawa 1993.
- [9] OSOWSKI S., *Sieci neuronowe do przetwarzania informacji*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2013.
- [10] YU H., WILAMOWSKI B. M., *Levenberg–Marquardt Training*, Ind. Electronics Handbook, Intelligent Systems, 2nd ed., 2011, Vol. 5, 12-1–12-15.
- [11] ORŁOWSKA-KOWALSKA T., KAMIŃSKI M., *FPGA Implementation of the Multilayer Neural Network for the Speed Estimation of the Two-Mass Drive System*, IEEE Transactions on Industrial Informatics, 2011, Vol. 7, No. 3, 436–445.
- [12] ŚWISULSKI D., *Cyfrowe metody rejestracji prędkości obrotowej maszyn wirujących*, Prace Naukowe Instytutu Maszyn, Napędów i Pomiarów Elektrycznych Politechniki Wrocławskiej, Nr 49, Seria: Studia i Materiały, Nr 21, 2000, 171–179.
- [13] QIAOYI L., YANLING X., WENLONG Y., JUNSHENG H., HUAN L., *Study on Color Analyzer based on the Multiplexing of TCS3200 Color Sensor and Microcontroller*, International Journal of Hybrid Information Technology, 2014, Vol. 7, No. 5, 167–174.
- [14] TAKAHASHI M., HASHIMUKAI H., ANDO H., *2-dimensional color sensor with combined neural network*, International Joint Conference on Neural Networks, 1991.

HARDWARE IMPLEMENTATION OF NEURAL DETECTOR FOR COLOR SELECTION

In this paper neural model applied in detection of wavelength of the light is presented. The structure of neural detector and design methodology are described. Moreover, hardware implementation is shown, for this purpose ATmega 32 is used. Tests presenting quality of detection has been prepared.