

Adam Głowacz*, Witold Głowacz*

Rozpoznawanie kształtów w sekwencjach wizyjnych z zastosowaniem algorytmu wstecznej propagacji błędów

1. Wprowadzenie

Obecnie istnieje wiele metod rozpoznawania kształtów [9, 15, 20, 22, 24]. Większość z nich jest oparta na przetwarzaniu danych [1, 3, 16–18]. Rozpoznawanie kształtów jest problemem złożonym [7, 12, 19]. Celem niniejszej pracy jest analiza systemu umożliwiającego rozpoznawanie kształtów w sekwencjach wizyjnych. Utworzono system, który jest modyfikacją systemu „Leaves Recognition v1.0” [25]. Zastosowano automatyczny mechanizm wprowadzania zdjęć i obróbki sekwencji wizyjnej. Przyjęto sieć neuronową opartą na algorytmie wstecznej propagacji błędów jako klasyfikator. Badania rozpoznawania kształtów zostały wykonane dla różnych sekwencji wizyjnych. System umożliwia wykonanie procesu trenowania i procesu identyfikacji kształtów na podstawie danych wejściowych i określonych przez użytkownika parametrów. Wyniki badań potwierdzają dużą skuteczność aplikacji jako systemu rozpoznawania kształtów w sekwencjach wizyjnych.

2. Opis systemu

System umożliwia wykonanie następujących funkcji:

- podziału sekwencji wizyjnej na ramki,
- wyznaczania obszaru pracy,
- rozpoznawanie kategorii w zależności od ustawień konfiguracji.

Po wykonaniu rozpoznawania kształtów wyniki są prezentowane na ekranie komputera lub są zapisywane do pliku.

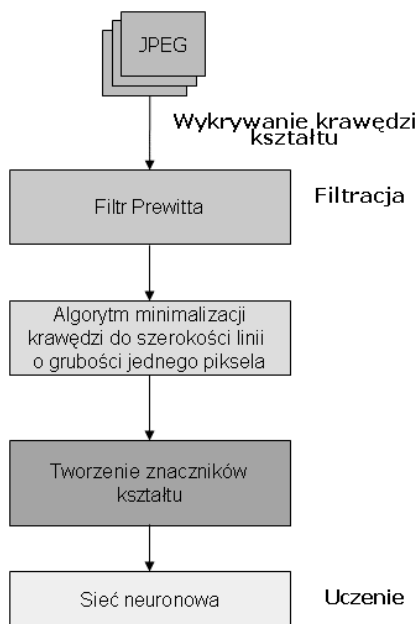
3. Proces rozpoznawania kształtów

Proces rozpoznawania kształtów składa się z procesu trenowania i procesu identyfikacji. Najpierw wykonywany jest proces trenowania [5]. Na początku procesu trenowania stosowane są algorytmy wykrywania krawędzi. Algorytmami wykrywania krawędzi są filtry

* Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

Prewitta i algorytm minimalizacji znalezionych krawędzi do szerokości linii o grubości jednego piksela. Następnie tworzone są znaczniki kształtu obrazu. Znaczniki są używane w uczeniu sieci neuronowej. Uczenie sieci neuronowej jest bardzo ważnym procesem, od którego zależą wyniki rozpoznawania. Następnie nowe próbki, które są nieznanne systemowi, używane są w procesie identyfikacji.

Proces trenowania jest przedstawiony na rysunku 1.

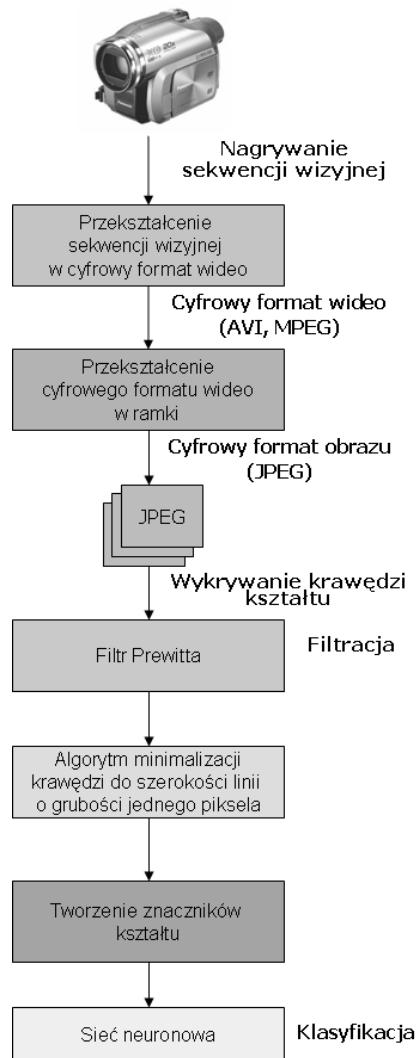


Rys. 1. Proces trenowania

W procesie trenowania wyróżnić można następujące kroki:

1. Filtracja.
2. Minimalizacja znalezionych krawędzi do szerokości linii o grubości jednego piksela.
3. Tworzenie znaczników kształtu obrazu.
4. Uczenie sieci neuronowej.

Nowa sekwencja wizyjna jest używana w procesie identyfikacji. Następnie sekwencja wizyjna przekształcana jest w cyfrowy format wideo (AVI – *Audio Video Interleave*, MPEG – *Moving Picture Experts Group*). Kolejnym krokiem jest przekształcenie cyfrowego formatu wideo w cyfrowy format obrazu (JPEG – *Joint Photographic Experts Group*). Kolejno stosowane są algorytmy wykrywania krawędzi kształtu (filtr Prewitta, algorytm minimalizacji znalezionych krawędzi do szerokości linii o grubości jednego piksela). Później są tworzone znaczniki kształtu obrazu. Sieć neuronowa oparta na algorytmie wstecznej propagacji błędów została zastosowana jako klasyfikator [13, 21]. Proces identyfikacji jest pokazany na rysunku 2.



Rys. 2. Proces identyfikacji

W procesie identyfikacji wyróżnić można następujące kroki:

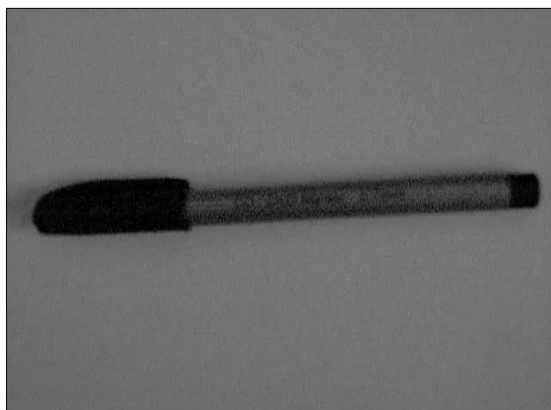
1. Nagrywanie sekwencji wizyjnej.
2. Przekształcenie sekwencji wizyjnej w cyfrowy format wideo (MPEG, AVI).
3. Przekształcenie cyfrowego formatu wideo w ramki (JPEG).
4. Filtracja.
5. Minimalizacja znalezionych krawędzi do szerokości linii o grubości jednego piksela.
6. Tworzenie znaczników kształtu obrazu.
7. Klasyfikacja siecią neuronową.

3.1. Nagrywanie sekwencji wizyjnej

Kamera cyfrowa nagrywa obraz wideo bardzo dobrej jakości. Nagrany obraz jest zapisany w cyfrowym formacie wideo. Z przenośnego nośnika danych obraz jest kopiowany na dysk komputera w celu przeprowadzenia właściwych obliczeń.

3.2. Przekształcanie cyfrowego formatu wideo w ramki

Cyfrowy format wideo jest przekształcany w ramki (JPEG) przez algorytmy kompresji. W tym celu zostały zastosowane skrypty w perlu i biblioteka mplayer. Zdjęcia mają rozdzielczość 640×480 pikseli (rys. 3).



Rys. 3. Zdjęcie długopisu przed obróbką

3.3. Filtracja

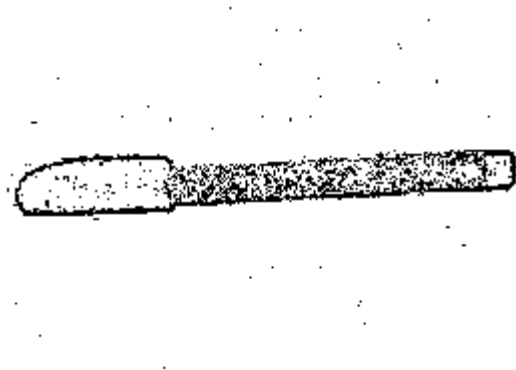
Filtry stosowane do analizy obrazów zakładają, że wykonywane na obrazie operacje będą kontekstowe. Oznacza to, że dla wyznaczenia jednego punktu obrazu wynikowego należy przeprowadzić określone obliczenia na wielu punktach obrazu źródłowego. Zadanie to polega na wyznaczeniu wartości funkcji, której argumentami są wartości piksela o tym samym położeniu na obrazie źródłowym oraz wartości pikseli z jego otoczenia K , które może mieć różną formę, ale najczęściej utożsamiane jest z kwadratowym oknem otaczającym symetrycznie aktualnie przetwarzany punkt obrazu. Z powodu kontekstowości wykonywanych operacji filtracja nie może dotyczyć pikseli znajdujących się bezpośrednio na brzegu obrazu. W komputerowej analizie obrazu dziedzina funkcji $L(m, n)$ jest dwuwymiarowa i nieciągła ponieważ taka jest reprezentacja obrazu. Konwolucję dla dwuwymiarowego dyskretnego obrazu można zapisać:

$$L(m, n) = (w \times L)(m, n) = \sum_{i, j \in K} L(m-i, n-j)w(i, j) \quad (1)$$

Filtry definiuje się jako tablice współczynników $w(i, j)$. Współczynniki $w(i, j)$ razem z odpowiednimi elementami obrazu $L(m-i, n-j)$ znajdującego się w oknie K ułożonym wokół punktu o współrzędnych m, n służą łącznie do obliczania wartości funkcji $L'(m, n)$ w danym punkcie na obrazie wynikowym. Punkt docelowy $L'(m, n)$ może być ustawiony w środku okna filtracji, a konwulacja będzie wyrażona wzorem:

$$\begin{aligned} L'(m, n) = & w_1 L(m-1, n-1) + w_2 L(m-1, n) + w_3 L(m-1, n+1) + \\ & + w_4 L(m, n-1) + w_5 L(m, n) + w_6 L(m, n+1) + \\ & + w_7 L(m+1, n-1) + w_8 L(m+1, n) + w_9 L(m+1, n+1) \end{aligned} \quad (2)$$

W przetwarzaniu obrazu używanych jest wiele metod [23]. W aplikacji zastosowano filtr Prewitta. Filtr ten tworzy obraz, w którym poszczególne odcienie szarości decydują o obecności krawędzi między obiektami. Filtr Prewitta jest zastosowany do wykrywania krawędzi (rys. 4).



Rys. 4. Zdjęcie długopisu po zastosowaniu filtru Prewitta

Działanie oparte jest na wykorzystaniu poziomego i pionowego filtru. Oba filtry są stosowane do obrazu. Kolejno wartości są sumowane do ostatecznego wyniku. Poziomy filtr Prewitta jest zdefiniowany za pomocą macierzy:

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (3)$$

Pionowy filtr Prewitta jest określony następująco:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (4)$$

Okno 3×3 zostało określone za pomocą macierzy:

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} \quad (5)$$

Moduł gradientu Prewitta jest zdefiniowany następująco:

$$Z = \sqrt{(X \cdot X) + (Y \cdot Y)} \quad (6)$$

gdzie:

$$X = -w_1 + w_3 - w_4 + w_6 - w_7 + w_9,$$

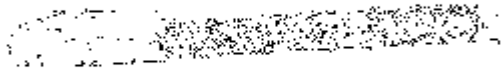
$$Y = w_1 + w_2 + w_3 - w_7 - w_8 - w_9,$$

Z – moduł gradientu Prewitta,

w_n – poziomy szarości piksela w oknie filtra z przedziału $[0, -255]$ dla $n = 1, 2, 3, 4, 5, 6, 7, 8, 9$.

3.4. Algorytm minimalizacji znalezionych krawędzi do szerokości linii o grubości jednego piksela

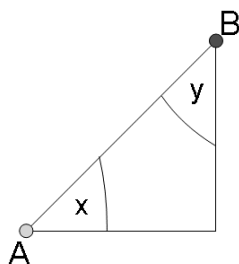
Aby poprawnie rozpoznawać kształty, należy rozpoznać zewnętrzną część krawędzi. Krawędź jest minimalizowana do szerokości linii o grubości jednego piksela. W tym celu zastosowany został algorytm, który minimalizuje znalezione krawędzie do szerokości linii o grubości jednego piksela (rys. 5).



Rys. 5. Zdjęcie długopisu po zastosowaniu algorytmu minimalizacji krawędzi do szerokości linii o grubości jednego piksela

3.5. Znaczniki kształtu obrazu

Znaczniki każdego kształtu są wyznaczone po wstępnym przetwarzaniu. Znaczniki są trójkątami prostokątnymi (rys. 6) [25].



Rys. 6. Trójkąt prostokątny, który reprezentuje znacznik kształtu obrazu

Kąty x i y są odpowiednie dla wejść sieci neuronowej. Dwa kąty reprezentują kierunek przeciwprostokątnej z punktu A do punktu B. Kosinusy i sinusy kątów x i y przy wierzchołkach A, B są użyte w reprezentacji kształtu obrazu (rys. 7). To jest bardzo ważne w rozpoznawaniu z zastosowaniem sieci neuronowej. Liczba znaczników zależy od zdjęcia i ustawień konfiguracji. Każda obliczona wartość jest podawana na wejścia sieci neuronowej.



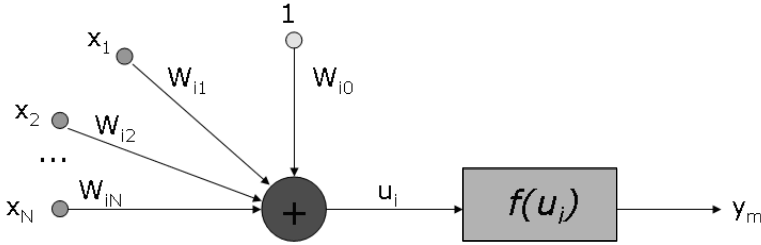
Rys. 7. Zdjęcie długopisu po zastosowaniu algorytmu tworzenia znaczników

3.6. Sieć neuronowa oparta na algorytmie wstecznej propagacji błędów

W sieci neuronowej został zastosowany algorytm wstecznej propagacji błędów. Na wejścia sieci neuronowej podane zostają wszystkie znaczniki kształtu. Każdy znacznik składa się z kosinusa i sinusa odpowiedniego kąta [25]. Liczba neuronów wejściowych jest dwukrotnie większa od wszystkich znaczników. Liczba neuronów wyjściowych odpowiada liczbie kategorii do rozpoznania. Różnica między kształtami polega na różnicach w krawędziach rozpatrywanych obrazów. Klasyfikacja używa znaczników i sieci neuronowej opartej na algorytmie wstecznej propagacji błędów (*backpropagation*).

Sieć neuronowa składa się z dużej liczby elementów przetwarzających informację. Elementy te nazywane są neuronami, chociaż w stosunku do rzeczywistych komórek nerwowych ich funkcje są bardzo uproszczone (rys. 8). Neurony są powiązane w sieć za pomocą połączeń o parametrach (tak zwanych wagach) modyfikowanych w trakcie procesu ucze-

nia [8]. Uczenie sieci neuronowej przy zastosowaniu algorytmu wstecznej propagacji błędów składa się z dwóch faz [4, 10, 11]. W pierwszej fazie są obliczane wartości neuronów. W drugiej fazie są modyfikowane wagi w zależności od błędów w każdej warstwie sieci [2, 6, 14].



Rys. 8. Model użytego neuronu. \mathbf{x} – wektor wejściowy, \mathbf{W}_i – wektor wag i -tego neuronu, y_m – wartość wyliczona przy zastosowaniu funkcji aktywacji, W_{i0} – próg

Wzór na wyliczenie wejścia u_i neuronu jest następujący:

$$u_i = \left(\sum_{j=1}^N W_{ji} x_j + W_{i0} \right) \quad (7)$$

Następnie wyznaczana jest wartość, która jest na wyjściu neuronu. Do tego celu należy użyć formuły:

$$y_m^{(j)} = f \left(\sum_{k \in M_i} w_k^{(m)(j)} y_k^{(j)} \right) \quad (8)$$

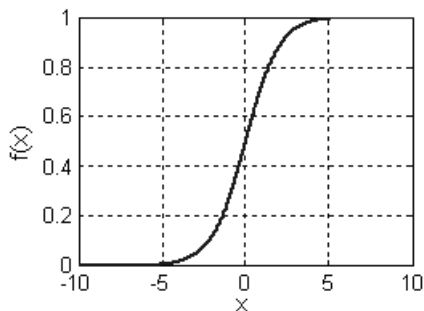
gdzie:

- M_i – zbiór numerów neuronów dostarczających sygnałów wejściowych do rozważanego aktualnie neuronu,
- $w_k^{(m)(j)}$ – wartość współczynnika wagowego synapsy łączącej wejście m -tego neuronu z wyjściem k -tego neuronu – podczas j -tego kroku uczenia,
- $f(x)$ – funkcja nieliniowa aktywacji (rys. 9).

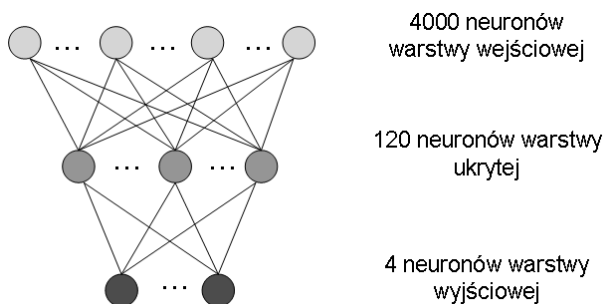
$$\delta_m^{(j)} = \sum_{k \in M} w_k^{(m)(j)} \delta_k^{(j)} \quad (9)$$

gdzie:

- M – zbiór neuronów, do których dociera sygnał wyjściowy z rozważanego neuronu warstwy ukrytej,
- $w_k^{(m)(j)}$ – wartość współczynnika wagowego synapsy łączącej wejście k -tego neuronu z wyjściem m -tego neuronu – podczas j -tego kroku uczenia,
- $\delta_m^{(j)}$ – błąd m -tego neuronu w warstwie ukrytej,
- $\delta_k^{(j)}$ – błąd k -tego neuronu warstwy wyjściowej, do którego dociera sygnał od neuronu m .



Rys. 9. Funkcja nieliniowa aktywacji



Rys. 10. Struktura sieci neuronowej zaimplementowana w aplikacji rozpoznawania kształtów

Strukturę sieci neuronowej zastosowaną w aplikacji przedstawia rysunek 10. Sieć uczy się na podstawie zaznaczonych znaczników kształtu obrazu. Po udanej nauce sieci neuronowej, wykonywana jest klasyfikacja. Obliczany jest błąd wyjściowej warstwy neuronów dla każdej kategorii. Skuteczność rozpoznawania obrazu jest zdefiniowana następująco:

$$I = 100 - \left(\frac{100T}{N} \right) \quad [\%] \quad (10)$$

gdzie:

- I – skuteczność rozpoznawania obrazu,
- N – liczba neuronów w warstwie wyjściowej,
- T – błąd warstwy wyjściowej.

Jeśli skuteczność rozpoznawania obrazu jest większa od 90%, to zakłada się, że kształt jest rozpoznany poprawnie.

Skuteczność rozpoznawania sekwencji wizyjnej jest zdefiniowana następująco:

$$R = \frac{100P}{A} \quad [\%] \quad (11)$$

gdzie:

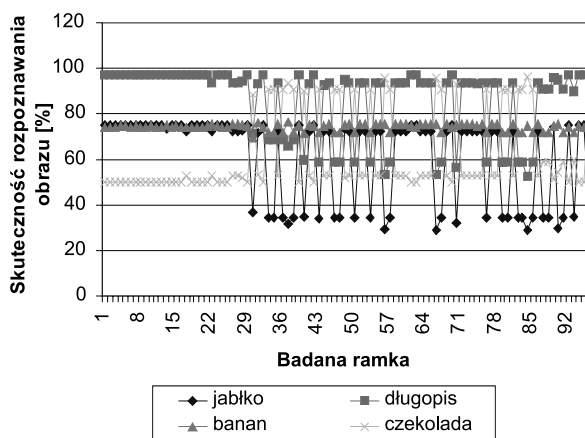
- R – skuteczność rozpoznawania,
- P – liczba prawidłowo rozpoznanych zdjęć,
- A – liczba wszystkich zdjęć.

4. Wyniki rozpoznawania kształtów

Rozpoznawanie kształtów zależy od wielu parametrów. Istotne jest, aby rozpoznawać kształty obrazów poprawnie. Największy wpływ na rozpoznawanie kształtów ma:

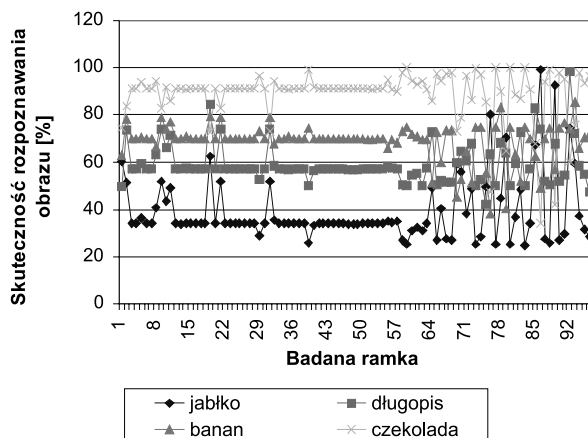
- jakość obrazu,
- parametry filtracji,
- parametry algorytmu tworzenia znaczników kształtu obrazu,
- parametry sieci neuronowej.

Badania zostały przeprowadzone dla określonych sekwencji wizyjnych z kształtem długopisu, czekolady, banana, jabłka. Wzorcowe zdjęcia kształtów zostały wykonane na jasnym tle. Sieć neuronowa uczyła się na 30 zdjęciach z każdej kategorii w procesie trenowania. Nowe nieznane systemowi obrazy uzyskane z sekwencji wizyjnej zostały użyte w identyfikacji. System miał określić kategorię, do której należy kształt w danej sekwencji. W procesie identyfikacji użyta została sekwencja wizyjna składająca się z 96 klatek. Na rysunkach 11–14 przedstawiono skuteczność rozpoznawania obrazu w zależności od badanej ramki sekwencji wizyjnej.



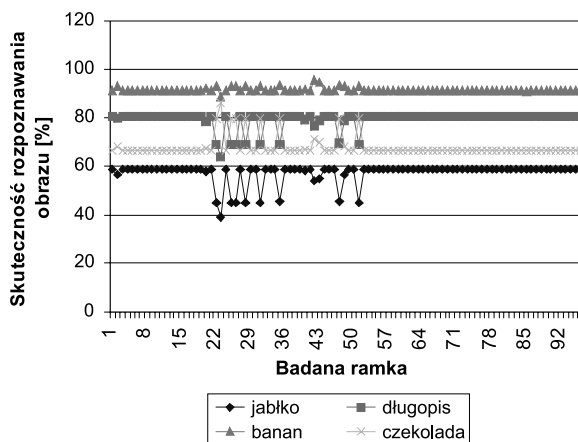
Rys. 11. Skuteczność rozpoznawania obrazu w zależności od badanej ramki sekwencji wizyjnej z kształtem długopisu

Pierwsze badania zostały wykonane dla sekwencji wizyjnej z kształtem długopisu. 72 z 96 ramek (zdjęć) zostało rozpoznanych poprawnie.



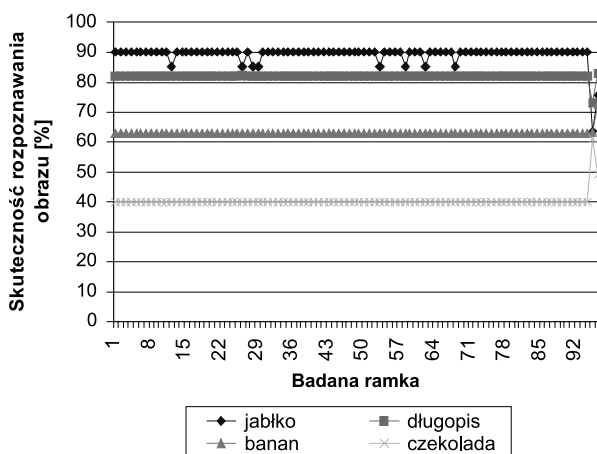
Rys. 12. Skuteczność rozpoznawania obrazu w zależności od badanej ramki sekwencji wizyjnej z kształtem czekolady

Badania zostały wykonane dla sekwencji wizyjnej z kształtem czekolady. 74 z 96 ramek zostało rozpoznanych poprawnie.



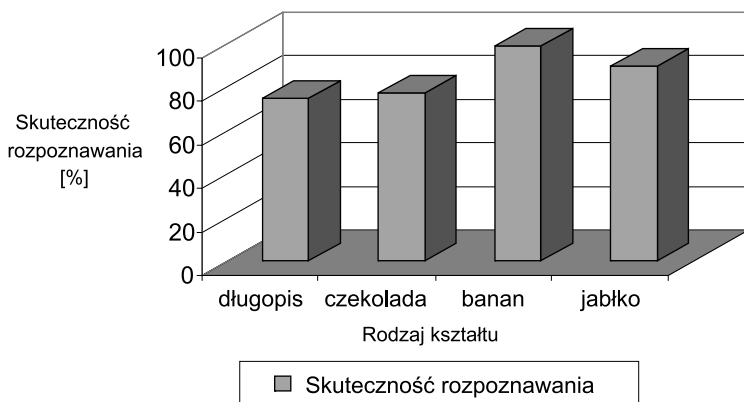
Rys. 13. Skuteczność rozpoznawania obrazu w zależności od badanej ramki sekwencji wizyjnej z kształtem banana

Następne badania przeprowadzono dla sekwencji wizyjnej z kształtem banana; 95 z 96 ramek z kształtem banana zostało rozpoznanych poprawnie.



Rys. 14. Skuteczność rozpoznawania obrazu w zależności od badanej ramki sekwencji wizyjnej z kształtem jabłka

Kolejne badania przeprowadzono dla sekwencji wizyjnej z kształtem jabłka; 86 z 96 ramek z kształtem jabłka zostało poprawnie rozpoznanych.



Rys. 15. Skuteczność rozpoznawania w zależności od rodzaju kształtu

Skuteczność rozpoznawania w zależności od rodzaju kształtu przedstawiona jest na rysunku 15. Skuteczność rozpoznawania dla sekwencji wizyjnej z kształtem długopisu wynosi 75%. Skuteczność rozpoznawania dla sekwencji wizyjnej z kształtem czekolady wynosi 77,08%, a skuteczność rozpoznawania dla sekwencji wizyjnej z kształtem jabłka wynosi 89,58%. Najlepsza skuteczność rozpoznawania dla sekwencji wizyjnej z kształtem banana jest równa 98,95%.

5. Podsumowanie

W celu automatycznego rozpoznawania sekwencji wizyjnych zmodyfikowano aplikację „Leaves Recognition v1.0”. Zastosowano automatyczny mechanizm wprowadzania zdjęć i obróbki sekwencji wizyjnej. Nowa aplikacja została przetestowana pod względem skuteczności rozpoznawania. Automatyczna aplikacja rozpoznawania kształtów identyfikuje kategorię poprawnie, jeśli skuteczność rozpoznawania obrazu jest powyżej 90%. Algorytmy przetwarzania danych i sztucznej inteligencji zostały zastosowane w procesie trenowania i identyfikacji. Analiza systemu pokazuje wrażliwość metody opartej na algorytmie wstecznej propagacji błędów. Badania ukazują, że sieć neuronowa pracuje dla różnych danych wejściowych. Najgorszą skuteczność rozpoznawania 75% uzyskano dla sekwencji wizyjnej z kształtem długopisu. Najlepszą skuteczność rozpoznawania 98,95% otrzymano dla sekwencji wizyjnej z kształtem banana. Czas procesu identyfikacji dla jednej próbki wynosi 0,989189 sekundy dla procesora Intel Pentium M 730.

Literatura

- [1] Ahn S.J., Rauh W., Recknagel M., *Geometric Fitting of Line, Plane, Circle, Sphere, and Ellipse*. Proc. of 6. ABW-Workshop Optische 3D-Formerfassung (Esslingen, Germany, 25–26 January, 1999), Technical Academy Esslingen, 1999.
- [2] Anderson J.A., *An Introduction to Neural Networks*. 1st ed., MIT Press, 1995.
- [3] Belongie S., Malik J., Puzicha J., *Shape Context: A New Descriptor for Shape Matching and Object Recognition*. Advances in Neural Information Processing Systems 13: Proc. 2000 Conf., T.K. Leen, T.G. Dietterich D.E., Tresp V., 2001, 831–837.
- [4] Chauvin Y., Rumelhart D.E., *Backpropagation: Theory, Architectures, and Applications*. Hillsdale, NJ: Erlbaum, 1995.
- [5] Cootes T., Cooper D., Taylor C., Graham J., *Active Shape Models – Their training and Application*. Computer Vision and Image Understanding (CVIU), vol. 61, no. 1, Jan. 1995, 38–59.
- [6] Fausett L.V., *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [7] Gavrilu D., Philomin V., *Real-Time Object Detection for Smart Vehicles*. Proc. Conf. Computer Vision, 1999, 87–93.
- [8] Girosi F., Jones M., Poggio T., *Regularization Theory and Neural Networks Architectures*. Neural Computation, vol. 7, no. 2, 1995, 219–269.
- [9] He Y., Kundu A., *2-D Shape Classification Using Hidden Markov Model*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, no. 11, Nov. 1991, 1172–1184.
- [10] Hirose Y., Yamashita K., Hijiya S., *Back-propagation algorithm which varies the number of hidden units*. Neural Networks, vol. 4, no. 1, 1991, 61–66.
- [11] Golden R.M., *Mathematical Methods for Neural Network analysis and Design*. MIT Press, 1996.
- [12] Johnson A.H., Hebert M., *Recognizing Objects by Matching Oriented Points*. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1997, 684–689.
- [13] Johnson A., Bron L., Brussee P., Irene de Goede, Hoogland S., *Learning from Mistakes*. Proceedings of Human System Interaction, 1–3 November 2000, Maastricht, Netherlands, 235–239
- [14] Kothari B., Paya B., Esat I., *Machinery fault diagnostics using direct encoding graph syntax for optimizing artificial neural network structure*. Proc. 1996 3rd Biennial Joint Conf. Engineering Systems Design and Analysis, ESDA, Part 7 (of 9). New York: ASME, 1996, 205–210.

- [15] Lanitis A., Taylor C.J., Cootes T.F., *A Generic System for Classifying Variable Objects Using Flexible Template Matching*. Proc. British Machine Vision Conf., vol. 1, 1993, 329–338.
- [16] Leung T., Malik J., *Contour continuity in region based image segmentation*. Lecture Notes in Computer Science, 1406, 1998, 544–559.
- [17] Liu H.-C., Srinath M.D., *Partial Shape Classification Using Contour Matching in Distance Transformation*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 11, Nov. 1990, 1072–1078.
- [18] Malik J., Belongie S., Leung T.K., Shi J., *Contour and Texture Analysis for Image Segmentation*. International Journal of Computer Vision, 43(1), 2001, 7–27.
- [19] Persoon E., Fu K., *Shape Discrimination Using Fourier Descriptors*. IEEE Trans. Systems, Man and Cybernetics, vol. 7, no. 3, Mar 1977, 170–179.
- [20] Prokop R.J., Reeves A.P., *A Survey of Moment-Based Techniques for Unoccluded Object Representation and Recognition*. CVGIP: Graphical Models and Image Processing, vol. 54, no. 5, pp. 438–460, Sept. 1992.
- [21] Skaf A., David B., Descotes-Genon B., Binder Z., *General approach to man-machine system design: ergonomic and technical specification of actions*. Proceedings of Human System Interaction, 1–3 November 2000, Maastricht, Netherlands, 355–367.
- [22] Sekita I., Kurita T., Otsu N., *Complex Autoregressive Model for Shape Recognition*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 14, no. 4, Apr. 1992, 489–496.
- [23] Shen J., Castan S., *An optimal linear operator for step edge detection*. Computer Vision, Graphics, and Image Processing: Graphical Models and Understanding, 54(2), 1992, 112–133.
- [24] Williams C.K., *Combining Deformable Models and Neural Networks for Hand-Printed Digit Recognition*. Dept. Computer Science, Univ. of Toronto, 1994 (PhD Thesis).
- [25] <http://www.jens-langner.de/lrecog/>