

## ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ МОВИ ЖЕСТИВ ЗА ДОПОМОГОЮ OPENCV, TENSORFLOW І KERAS РЕАЛІЗОВАНИХ У PYTHON

*Безверхий О.І.*, доктор фізико-математичних наук, Національний транспортний університет, Київ, Україна, o\_bezver@ukr.net, orcid.org/0000-0002-0834-6335

*Луц В.Є.*, Національний транспортний університет, Київ, Україна, tibet.septim@gmail.com, orcid.org/0009-0001-2948-6935

### DESIGN OF A SIGN LANGUAGE RECOGNITION INFORMATION SYSTEM USING OPENCV, TENSORFLOW AND KERAS IMPLEMENTED IN PYTHON

*Bezverkhyi O.I.*, Doctor of physical and mathematical Science, National transport university, Kyiv, Ukraine, o\_bezver@ukr.net, orcid.org/0000-0002-0834-6335

*Luz V.E.*, National transport university, Kyiv, Ukraine, tibet.septim@gmail.com, orcid.org/0009-0001-2948-6935

#### Вступ

Проблема розробки мультимедійних систем розпізнавання елементів дактильної абетки жестової мови, як одного із засобів комунікації між людьми з вадами слуху, – актуальна [1]. Аналіз сучасних систем розпізнавання дактильної жестової мови для систем сурдоперекладу представлено в роботі [2]. Останніми роками прогрес у галузі штучного інтелекту (ШІ) і, зокрема, нейронних мереж, відкрив нові можливості для покращення життя людей з обмеженими можливостями. Однією з сфер, де нейронні мережі зробили значний внесок, є розпізнавання мови жестів.

Нейронні мережі – це підмножина алгоритмів машинного навчання, створених на основі нейронної структури людського мозку. Ці мережі вивчають шаблони та зв'язки з даних, щоб робити прогнози чи класифікації. Використовуючи нейронні мережі, ми можемо розробляти програми, які задовольняють унікальні потреби людей з обмеженими можливостями, допомагають їм подолати комунікаційні бар'єри та сприяють інклюзивності.

В роботах Крака Ю.В. [3,4,5] описується проблема адаптації людей з вадами слуху, а також описується процес створення інформаційної системи для людей з вадами слуху використовуючи сучасні технології нейромереж. З метою виділення ефективних характеристик представлення кисті руки при відтворенні дактилем було проведено досліді [4]. В роботі [5] створюється інформаційна система для розпізнавання жестів української дактильної абетки за допомогою кросплатформених технологій. Автори використовують машинне навчання, глибоке навчання, а також створенням моделі тривимірного скелету руки для розпізнавання жестів за допомогою тривимірних згорток.

Мета даної роботи – проектування інформаційної системи розпізнавання української мови жестів за допомогою Python.

#### Система розпізнавання мови жестів

Розпізнавання мови жестів [6]. Мова жестів (рис.1) – це візуальна мова, яка використовує комбінацію жестів рук, міміки та рухів тіла для передачі сенсу. Розпізнавання мови жестів передбачає розробку системи, здатної інтерпретувати та розуміти жести мовою жестів.

Для проектування інформаційної системи розпізнавання мови жестів, ми виконаємо такі кроки:

##### 1. Збір даних:

– Збір наборів даних жестів, включаючи відеозаписи різних знаків, які виконуються різними підписувачами.

– Додавання анотації до набору даних, пов'язуючи кожне відео з відповідною міткою мовою жестів.

##### 2. Попередня обробка даних:

– Перетворення відеозаписів на кадри або послідовності кадрів.

– Застосовання методу обробки зображень для покращення якості кадрів, такі як зміна розміру, нормалізація та зменшення шуму.



Рисунок 1 – Дактильна абетка української мови  
Figure 1 – Dactylic alphabet of the Ukrainian language

### 3. Навчання нейронної мережі:

- Розділення набору даних на набори для навчання та тестування, щоб оцінити продуктивність моделі.
- Створення архітектури нейронної мережі, придатну для розпізнавання мови жестів, таку як згорточна нейронна мережа (CNN) або рекурентна нейронна мережа (RNN).
- Навчання мережі за допомогою навчального набору даних, регулюючи ваги мережі, щоб мінімізувати помилку між прогнозованими та фактичними мітками.
- Оптимізація гіперпараметрів мережі, таких як швидкість навчання та розмір пакету, для підвищення продуктивності.

### 4. Оцінка моделі:

- Оцінювання навченої моделі на наборі даних тестування для оцінювання її точності і узагальненості.
- Використання показників оцінки, такі як точність, запам'ятовування та оцінка F1 для вимірювання ефективності моделі.

### 5. Розгортання:

- Збереження навченої моделі як файл для подальшого використання.
- Розроблення програми Python, яка завантажує навчену модель і приймає відеовхід для розпізнавання жестів мовою жестів у реальному часі.
- Використовування бібліотек OpenCV і TensorFlow/Keras для обробки відео та операцій нейронної мережі.

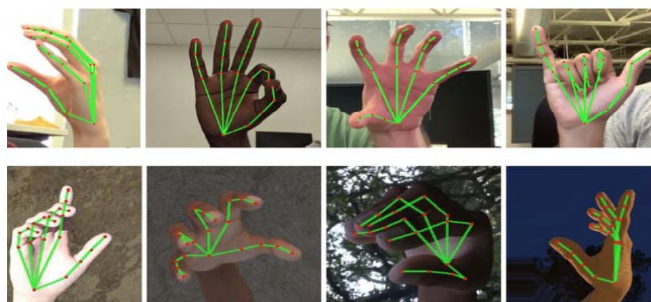


Рисунок 2 – Приклад демонстрації роботи нейромережі для розпізнавання долоні(TensorFlow)  
Figure 2 – An example of a demonstration of a neural network for palm recognition (TensorFlow)

### Технології використання нейромереж для мови жестів

Мова жестів є найважливішим засобом спілкування для людей з вадами слуху. Однак розуміння та переклад мови жестів може бути проблемою для тих, хто не має необхідної підготовки чи досвіду. Останніми роками досягнення комп'ютерного зору та машинного навчання відкрили нові можливості для систем розпізнавання мови жестів, які можуть допомогти людям з обмеженими можливостями ефективно спілкуватися. У цій статті розглядається інтеграція OpenCV, TensorFlow і Keras, реалізованих у Python, для розробки надійної та точної системи розпізнавання мови жестів.

Розуміння OpenCV:

OpenCV[7] (Бібліотека комп'ютерного бачення з відкритим кодом) – це потужна бібліотека з відкритим кодом, яка надає широкий спектр алгоритмів та інструментів комп'ютерного зору. Він пропонує підтримку обробки зображень і відео, виявлення об'єктів, виділення функцій тощо. OpenCV є фундаментальним компонентом у нашій системі розпізнавання мови жестів, оскільки дає нам змогу попередньо обробляти й аналізувати зображення або відеокадри мовою жестів для виділення значущих функцій.

Приклад коду з використання бібліотеки OpenCV. Завантаження та відображення зображення за допомогою OpenCV:

```
import cv2

# Завантаження зображення з файлу
image = cv2.imread('sign_language.jpg')

# Демонстрація зображення
cv2.imshow('Sign Language', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Використання TensorFlow і Keras:

TensorFlow[8,9], фреймворк машинного навчання з відкритим вихідним кодом, розроблений Google, разом із високорівневим API Keras надає комплексну платформу для створення та навчання нейронних мереж. Ці інфраструктури дозволяють нам створювати та навчати складні моделі, здатні розпізнавати жести мовою жестів із надзвичайною точністю. TensorFlow надає основний обчислювальний графік і методи оптимізації, а Keras пропонує зручний інтерфейс для визначення та навчання моделей глибокого навчання.

Приклад коду з використанням бібліотеки TensorFlow. Побудова згорткової нейронної мережі (CNN) у TensorFlow:

```
import tensorflow as tf
from tensorflow.keras import layers

# Визначте архітектуру моделі CN
model = tf.keras.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(num_classes, activation='softmax'))

# Скомпілювання моделі
model.compile(optimizer='adam',
loss=tf.keras.losses.SparseCategoricalCrossentropy(),
metrics=['accuracy'])
```

Keras[9] – це API високого рівня, створений на основі TensorFlow, який спрощує процес створення та навчання моделей глибокого навчання. Він пропонує інтуїтивно зрозумілий і зручний інтерфейс для визначення нейронних мереж, що дозволяє розробникам створювати складні архітектури за допомогою лише кількох рядків коду. Keras підтримує різні типи шарів, функції втрати та оптимізатори, що робить його універсальним інструментом для розробки моделей розпізнавання мови жестів.

Навчання моделі розпізнавання мови жестів за допомогою Keras:

```
import tensorflow as tf
from tensorflow import keras
```

```

# Завантажте набір даних розпізнавання мови жестів
(X_train, y_train), (X_test, y_test) = keras.datasets.mnist.load_data()

# Попередня обробка даних (наприклад, нормалізація значень пікселів)
X_train = X_train / 255.0
X_test = X_test / 255.0

# Визначення архітектури моделі
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])

# Компілювання моделі
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Тренування моделі
model.fit(X_train, y_train, epochs=10)

# Оцінка моделі
test_loss, test_acc = model.evaluate(X_test, y_test)
print('Test accuracy:', test_acc)

```

Ці приклади дають уявлення про те, як OpenCV, TensorFlow і Keras можна використовувати для розпізнавання мови жестів у Python [10,11]. Однак для комплексної системи розпізнавання мови жестів потрібні додаткові кроки, такі як попередня обробка даних, навчання моделі на наборах даних мови жестів і обробка в реальному часі. Тим не менш, ці фреймворки пропонують міцну основу для створення та впровадження таких систем.

### **Створення інформаційної системи для розпізнавання мови жестів за допомогою OpenCV, TensorFlow і Keras у Python**

Детальне та покрокове пояснення програмного коду:

1. Імпортуйте необхідні бібліотеки, включаючи TensorFlow, Keras, OpenCV і NumPy.
2. Встановіть шлях до набору даних мови жестів і визначте список позначок мови жестів.
3. Ініціалізуйте порожні списки для зберігання даних і міток.
4. Завантажте та попередньо обробіть набір даних, перебираючи каталоги міток, завантажуючи кожне зображення, змінюючи його розмір до фіксованого розміру (наприклад, 64x64 пікселів) і додавання його до списку даних. Відповідна мітка додається до цільового списку.
5. Перетворіть дані та цільові списки на масиви NumPy.
6. Розділіть набір даних на набори для навчання та тестування за допомогою функції `train\_test\_split` від scikit-learn.
7. Перетворіть мітки на вектори з одноразовим кодуванням за допомогою функції `to\_categorical` від Keras.
8. Визначте модель розпізнавання мови жестів за допомогою Sequential API від Keras. У цьому прикладі використовується проста архітектура CNN із згортковими шарами та шарами об'єднання, за якими слідує повністю зв'язані шари.
9. Скомпілюйте модель, вказавши оптимізатор, функцію втрат і метрику оцінки.
10. Навчіть модель за допомогою навчальних даних і перевірте її за допомогою даних тестування.
11. Оцініть навчену модель на основі даних тестування та роздрукуйте точність тесту.
12. Збережіть навчену модель у файл методом `save`.
13. Визначте функцію `recognize\_sign\_language`, яка приймає вхідне зображення, змінює його розмір, нормалізує його та передбачає мітку мови жестів за допомогою навченої моделі.

14. Ви можете використовувати цю функцію для розпізнавання жестів мовою жестів, передаючи їй зображення.

Даний код виконує функцію тренування нейромережі для розпізнавання жестів по алфавіту.

```
import os
import numpy as np
import cv2
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from keras.utils import to_categorical

# Set the path to your sign language dataset
dataset_path = '/path/to/your/dataset'

# Define a list of sign language labels
labels = ['A', 'Б', 'В', 'Г', 'Д', 'Е', 'Є', 'Ж', 'З', 'Т', 'Ї', 'Й', 'К', 'Л', 'М', 'Н', 'О']

# Initialize empty lists to store the data and labels
data = []
target = []

# Load the dataset and preprocess the data
for label in labels:
    label_path = os.path.join(dataset_path, label)
    for image_file in os.listdir(label_path):
        image_path = os.path.join(label_path, image_file)
        image = cv2.imread(image_path)
        image = cv2.resize(image, (64, 64)) # Resize the image to a fixed size
        data.append(image)
        target.append(label)

# Convert data and target to NumPy arrays
data = np.array(data, dtype="float32")
target = np.array(target)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data, target, test_size=0.2, random_state=42)

# Convert the labels to one-hot encoded vectors
num_classes = len(labels)
y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)

# Define the sign language recognition model
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

```

# Compile and train the model
model.compile(optimizer='adam',loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))

# Evaluate the model
_, accuracy = model.evaluate(X_test, y_test)
print('Test Accuracy:', accuracy)

# Save the trained model
model.save('sign_language_model.h5')

# Example usage of the trained model for sign language recognition
def recognize_sign_language(image):
    image = cv2.resize(image, (64, 64)) # Resize the input image
    image = np.expand_dims(image, axis=0) # Add an extra dimension for batch size
    image = image.astype('float32') / 255 # Normalize the image
    predicted_vector = model.predict(image)
    predicted_label = labels[np.argmax(predicted_vector)]
    return predicted_label

```

## Висновки

У результаті виконання роботи було:

1. Проаналізовано сучасні технології створення систем штучного інтелекту.
2. Досліджено сучасні бібліотеки для розпізнавання мови жестів, такі як TensorFlow, Keras і OpenCV.
3. Наведені практичні приклади використання бібліотек для нейромереж використовуючи мову програмування Python.
4. Створений програмний код для тренування нейромережі розпізнавати мови жестів, а точніше жести алфавіту.

Нейронні мережі пропонують потужний підхід до покращення доступності для людей з обмеженими можливостями, особливо в області розпізнавання мови жестів. Використовуючи можливості штучного інтелекту, ми можемо розробляти надійні та точні системи, які усувають комунікаційні прогалини та сприяють інклюзивності. Python зі своєю багатою екосистемою бібліотек і фреймворків забезпечує універсальну платформу для створення та розгортання таких систем. Розширення можливостей людей з обмеженими можливостями за допомогою технології ШІ є значним кроком до створення більш інклюзивного суспільства.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Кульбіда С.В. Українська дактилологія: Наук.-метод. посібник. – К.: Педагогічна думка, 2007. – 256 с.
2. Аналіз сучасних систем розпізнавання дактильної жестової мови для систем сурдоперекладу/ Лихошерстов Д.О. Лебедев Д.Ю.// Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки. Том 32 (71) № 6 2021, С.44-48.
3. Информационная технология идентификации дактилем украинского жестового языка / Ю. В. Крак, А. В. Бармак, В. С. Касьянюк, Д. В. Шкильнюк // Управляющие системы и машины. – 2015. – № 6. – С. 23–28.
4. Інформаційна технологія для моделювання української мови жестів / Ю.Г. Кривонос, Ю.В. Крак, О.В. Бармак та ін. // Штучний інтелект. – 2009. – № 3. – С. 186–197.
5. С.С. Кондратюк. Платформонезалежне програмне забезпечення для розробки систем жестової комунікації: моделювання дактильної мови / С.С.Кондратюк, Ю.В. Крак // Штучний інтелект.– т.73 в.3 – 2016. С.36-47.
6. Жестова мова, як повноцінний засіб комунікації. Режим доступу: <https://naurok.com.ua/zhestova-mova-yak-povnocinniy-zasib-komunikaci-332294.html>

7. OpenCV – бібліотека для розпізнавання об’єктів у зображеннях та камерах. Режим доступу: <https://blog.desdelinux.net/uk/opencv-бібліотека-для-розпізнавання-об%27єктів-у-зображеннях-та-камерах/>

8. TensorFlow Core. Documentation. Режим доступу: [https://www.tensorflow.org/api\\_docs/python/tf](https://www.tensorflow.org/api_docs/python/tf)

9. TensorFlow Core. Keras. Режим доступу: [https://www.tensorflow.org/guide/keras/sequential\\_model](https://www.tensorflow.org/guide/keras/sequential_model)

10. OpenCV-Python Tutorial. Режим доступу: [https://docs.opencv.org/3.4/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html)

11. TensorFlow for Beginners With Examples and Python Implementation. Режим доступу: <https://www.analyticsvidhya.com/blog/2021/11/tensorflow-for-beginners-with-examples-and-python-implementation/>

## REFERENCES

1. Kulbida S.V. Ukrainian dactylogy: Scientific method manual. – К.: Pedagogical thought, 2007. – 256 p.

2. Analysis of modern dactyl sign language recognition systems for sign language translation systems/ Lykshosherstov D.O. Lebedev D.Yu.// Scientific notes of TNU named after V.I. Vernadskyi. Series: Technical sciences. Volume 32 (71) No. 6 2021, P.44-48.

3. Information technology of dactyl identification of Ukrainian sign language / Yu. V. Krak, O. V. Barmak, V. S. Kasyanyuk, D. V. Shkilnyuk // Control systems and machines. – 2015. – No. 6. – P. 23–28.

4. Information technology for modeling Ukrainian sign language / Yu.G. Kryvonis, Yu.V. Krak, O.V. Barmak et al. // Artificial Intelligence. – 2009. – No. 3. – P. 186-197.

5. S.S. Kondratyuk. Platform-independent software for the development of gestural communication systems: modeling of dactyl speech / S.S. Kondratyuk, Yu.V. Krak // Artificial intelligence. – Vol. 73, Art. 3 – 2016. P. 36-47.

6. Sign language as a full-fledged means of communication. Access mode: <https://www.naurok.com.ua/zhestova-mova-yak-povnocinniy-zasib-komunikaci-332294.html>

7. OpenCV is a library for recognizing objects in images and cameras. Access mode: <https://blog.desdelinux.net/uk/opencv-library-for-object-recognition-in-images-and-cameras/>

8. TensorFlow Core. Documentation. Access mode: [https://www.tensorflow.org/api\\_docs/python/tf](https://www.tensorflow.org/api_docs/python/tf)

9. TensorFlow Core. Keras. Access mode: [https://www.tensorflow.org/guide/keras/sequential\\_model](https://www.tensorflow.org/guide/keras/sequential_model)

10. OpenCV-Python Tutorial. Access mode: [https://docs.opencv.org/3.4/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html)

11. TensorFlow for Beginners With Examples and Python Implementation. Access mode: <https://www.analyticsvidhya.com/blog/2021/11/tensorflow-for-beginners-with-examples-and-python-implementation/>

## РЕФЕРАТ

Безверхий О.І. Проектування інформаційної системи розпізнавання мови жестів за допомогою OpenCV, TensorFlow і Keras реалізованих у Python / О.І. Безверхий, В.Є. Луц // Вісник Національного транспортного університету. Серія «Технічні науки». Науковий, науково-виробничий журнал. – К.: НТУ, 2023. – Вип. 3 (57).

Жестова мова є одним із основних засобів передачі інформації та спілкування, разом із текстом і мовленням. Жестові мови складаються з окремих знаків, які поєднуються в букви, слова, фрази за допомогою послідовного переходу від одного знаку до іншого. У статті розглянуто і розроблено складові інформаційної технології перекладу української жестової мови. Об’єктом дослідження є процес розпізнавання та моделювання дактилем української дактильної мови. Предметом дослідження є методи аналізу зображень дактилем та процесу розпізнавання та моделювання дактилем. Метою даної роботи є проектування інформаційної системи розпізнавання української мови жестів за допомогою Python. У цій статті розглядається інтеграція OpenCV, TensorFlow і Keras, реалізованих у Python, для розробки надійної та точної системи розпізнавання мови жестів. Наведено детальне та покрокове пояснення програмного коду створення інформаційної системи для розпізнавання мови жестів за допомогою OpenCV, TensorFlow і Keras у Python.

Створені у процесі дослідження програмні засоби опрацювання та перекладу української жестової мови можна використати під час проектування інформаційної системи перекладу з однієї мови на іншу, що полегшить спілкування осіб із вадами слуху між собою і з тими, хто не володіє жестовою мовою.

КЛЮЧОВІ СЛОВА: ЖЕСТОВА МОВА, НЕЙРОННІ МЕРЕЖІ, ІНФОРМАЦІЙНА СИСТЕМА PYTHON, ДАКТИЛЬНА АБЕТКА.

#### ABSTRACT

Bezverhyi O.I., Luz V.S. Design of a sign language recognition information system using OpenCV, TensorFlow and Keras implemented in Python. Visnyk National Transport University. Series «Technical sciences». Scientific, scientific and industrial journal. – К.: NTU, 2023. – Issue 3 (57).

Sign language is one of the main means of information transfer and communication, along with text and speech. Sign languages consist of individual signs that are combined into letters, words, and phrases using a sequential transition from one sign to another. The article examines and develops the components of Ukrainian sign language translation information technology. The object of the study is the process of dactyl recognition and modeling of the Ukrainian dactyl language. The subject of research is the methods of dactyl image analysis and the process of dactyl recognition and modeling. The purpose of this work is to design an information system for recognizing Ukrainian sign language using Python. This paper examines the integration of OpenCV, TensorFlow, and Keras implemented in Python to develop a robust and accurate sign language recognition system. A detailed and step-by-step explanation of the software code for creating an information system for sign language recognition using OpenCV, TensorFlow and Keras in Python is given

The software tools for studying and translating Ukrainian sign language created during the research can be used when designing an information system for translation from one language to another, which will facilitate communication between people with hearing impairments and with those who do not know sign language.

KEYWORDS: SIGN LANGUAGE, NEURAL NETWORKS, PYTHON INFORMATION SYSTEM, DACTILE ALPHABET.

#### АВТОРИ:

Безверхий Олександр Ігорович, д.т.н., професор, Національний транспортний університет, Київ, Україна, e-mail: o\_bezver@ukr.net.

Луц Владислав Євгенович, Національний транспортний університет, Київ, Україна, e-mail: tibet.septim@gmail.com.

#### AUTHORS:

Bezverkhyy Oleksandr Ihorovych, Doctor of Technical Sciences, professor, National Transport University, Kyiv, Ukraine, e-mail: o\_bezver@ukr.net.

Luz Vladyslav Evgenovich, National Transport University, Kyiv, Ukraine, e-mail: tibet.septim@gmail.com.

#### РЕЦЕНЗЕНТИ:

Гавриленко Валерій Володимирович., д-ф.м-н., професор, Національний транспортний університет, Київ, Україна.

Івохін Євген Вікторівич., д-ф.м-н., професор, Національний університет імені Тараса Шевченка, Київ, Україна.

#### REVIEWERS:

Gavrilenko Valeriy Volodymyrovych, Doctor of Physical and Mathematical sciences, professor, National Transport University, Kyiv, Ukraine.

Ivohin Evgen Viktorovych, Doctor of Physical and Mathematical sciences, professor, National university Taras Shevchenko, Kyiv, Ukraine.