| **Project Proposal** | | | | |
| --- | --- | --- | --- | --- |
| 교과목명 | 창직종합설계프로젝트1 PROJECT1 | | 교과목번호 | 101811 - 4 |
| 학점(설계)/시수 | 3 / Mon(10), Wed(10), Thu(10) | | | |
| 개설학과(학년) | 4 | | 담당교수 | Junyong Lee |
| 과제명 | Using OpenCv and Kafca to recognize faces and recommend advertisements. | | | |
| 팀원 | 학번 | 이름 | | 연락처/e-mail |
| B711069 | 박선형 | | 010-5026-1897/  byjs0908@naver.com |
| B711134 | 이승환 | | 010-9135-4481/  lezend22@gmail.com |
| B711211 | 허민욱 | | 010-8734-2429/  hmw1022@naver.com |
|  |  | |  |

**1. OverView**

Use OpenCv to collect information about users who use the program. Create a machine learning model to process the collected data and learn the collected facial data values. The collected data is parsed through the machine on which it was produced. Send feature values derived from machine learning models to middleware as subscription topic values.

Middleware compares the stored topic with the value of the advertisement data’s tag delivered from the publisher and sends the advertisement data to the client(consumer) who subscribed to the topic.  
In addition, kafka is used for effective data distribution processing in the middleware.Use OpenCv to collect information about users who use the program. Create a machine learning model to process the collected data and learn the collected facial data values. The collected data is parsed through the machine on which it was produced. Send feature values derived from machine learning models to middleware as subscription topic values. It sends the topic provided by the subscriber to the middleware, and the middleware goes through the process of comparing the topic with the information it has. If there is a value that matches Topic, Kafca returns that value to the client.

**2. necessity and motivation**

- analysis of existing products or services

In the ever-expanding big data market, the importance of face data identifying individuals is increasing. Until now, the use of facial data is expected to be indispensable for society in the future, considering that facial data information gives the elderly and the digital underprivileged high access to services and can be used for various social-based services in the future. In order to preemptively respond to these socio-technical demands, we aim to launch a service based on face recognition. As a schematic topic, it aims to establish topic-type pub/sub middleware that recommends advertisements suitable for objects by analyzing and classifying characteristic points of objects as ML models based on face recognition. In the world of big data, automated feature point classification through machine learning will be established to effectively select and supply data for each object from among the pouring data, and performance evaluation will be conducted to run not only one client but also a lot of data.

**3. key technology**

OpenCV for face recognition.

ML tensorflow.keras for face data learning models.  
Use of Kafka for building Pub/Sub middleware and effectively distributing asynchronous data.

**4. details**

**[**Facial recognition stage**]**

OpenCv, one of the image recognition libraries, is used to utilize the user's face data.In this case, faces of various people are recognized through the camera, and face data obtained therefrom is separated into frame units.

After that, OpenCv extracts matrix values from each frame unit using a haar cascade recorder.

**[**design the layer of the tensorflow.keras ML stage**]**

The matrix values obtained by the above method are trained in the basic ML model designed.In this way, when the basic ML model is learned, the learned ML model comes out as a result value. At this time, the user inputs his or her face data into this learned ML model and passes it to receive feature values as result values.At this time, the feature value includes age, gender, and emotion.Thereafter, the user transmits the feature value to the middleware as subscription information (topic).

**[**middleware construction stage**]**

To make pub/sub Frame, make producers (server, microservice, application), Consumer (program), and MiddleWare respectively.At this time, all microservices are managed within the cluster using kafka.

At this time, you can also create an object to implement pub/sub Frame.

kafka functions similarly to the messaging queue. In this case, the user (consumer) transmits the topic to MiddleWare in the messaging queue method.MiddleWare saves the user in {ip, port : topic} form and sends an ack signal to Consumer to let them know that the delivery has been completed without any problems. After that, producer produces ads and supplies information to MiddleWare as follows: {#20s, #soccer, #url} In this case, Middleware compares the topic(tag) in the advertisement with the subscribed topic information and sends all the information it has received so far if the value matches.

**[**Distributed data processing stage**]**

As the number of partitions or consumer groups that store Topic increases, performance can deteriorate rapidly, and at this time, distributed data processing is used to prevent performance degradation.

**5.requirement**

1. The application of the microservice architecture ensures broad scalability and low combination: what is needed to be implemented as middleware that can cover a wide range of sectors. be essential for platform application

2. Security: Store the characteristic value (topic) as encrypted information without storing it as original data so that individual characteristics cannot be identified. The security required is also clear as it deals with sensitive individual data. Prevent brokers in middleware from looking into personal original data.

3. Asynchronous: The role of middleware that mediates between multiple client servers and publisher servers ensures asynchronous connectivity, reducing the burden of network communication.

4. Appropriate advertising recommendation system: Analyzing topics through recommended algorithms and transmitting highly relevant advertising data to the client.

**6.System Design (Composition and Analysis)**

Consumer Client: Learn features using ML deep learning model by extracting facial data from individuals.In addition, asynchronous communication is performed through an RPC server and a server that sends feature points to middleware as topics.

MessageMiddleWare: When receiving a topic, check your registration via ack message. When topic matching, publisher-generated vast amounts of advertising data are transferred to the corresponding server address (ip, port) using the recommended algorithm.

Publisher Client: Send advertising tags via asynchronous communication with middleware. Sent as a transmission message in the form of advertising (url, #tag) text.

**7. Embody**

making JVM-based language (Java)/Python3-based middleware and client

Use pub/sub architecture through Kafka framework, adjust broker through zipper

Extract facial data through OpenCv library

Proceed with ML deep learning through the TensorFlow.keras library.

asynchronous communication via gRpc

**8. Test**

**[**face recognition part**]**   
It constitutes a dataset to successfully extract feature points through diversified facial photographs of objects.Subsequently, the threshold of the ML model that successfully extracts the feature points was determined through a test.

**[**client-middleware communication**]**   
Check Rpc server call between servers, check if inter-server communication is possible with ack message log output.

**[**confirmation of distributed processing of passive data**]**   
A test set is produced to evaluate the performance of distributed processing of about 1 million advertisement data. Adjust the topics (partition groups) in the broker to derive experimentally possible levels.

**[**security verification**]**   
Tests security by checking whether the original data of individuals in the broker is transmitted and also checking whether there is a possibility of reverse tracking if encrypted.

**9. roles of team members**

박선형 : implementation of rpc server, consumer client.

이승환 : Face recognition code, face data feature point extraction ML model design, middleware production

허민욱 : middleware implementation, publisher client implementation, test set production.

**10.Schedule**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 주  설계요소 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| planning proposal | o | o | o |  |  |  |  |  |  |  |  |  |  |  |  |  |
| data survey | o | o | o |  |  |  |  |  |  |  |  |  |  |  |  |  |
| reusable tech research |  | o | o |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ML model design |  |  |  | o | o | o | o |  |  |  |  |  |  |  |  |  |
| WAS production |  |  |  | o | o | o | o |  |  |  |  |  |  |  |  |  |
| model-applied middleware development |  |  |  |  |  |  | o | o | o |  |  |  |  |  |  |  |
| middleware development |  |  |  |  |  |  | o | o | o |  |  |  |  |  |  |  |
| Apache Kafka application |  |  |  |  |  |  | o | o | o |  |  |  |  |  |  |  |
| Service Applied Distribution |  |  |  |  |  |  |  |  | o | o | o |  |  |  |  |  |
| confirmation of practical use of massive data |  |  |  |  |  |  |  |  | o | o | o | o |  |  |  |  |
| Performance evaluation and correction of performance improvement points |  |  |  |  |  |  |  |  |  |  |  | o | o | o | o | o |
| Preparation of report and preparation for graduation exhibition |  |  |  |  |  |  |  |  |  |  |  |  |  | o | o | o |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |