

**МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ**  
**ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААН, ИНЖЕНЕРЧЛЭЛИЙН СУРГУУЛЬ**  
**МЭДЭЭЛЭЛ, КОМПЬЮТЕРИЙН УХААНЫ ТЭНХИМ**

Цэрэнбямбаагийн Хасбилэгт

**ГАР БИЧМЭЛ ТАНИЛТАНД ЗОРИУЛСАН**  
**МУИС-Н НЭЭЛТТЭЙ ӨГӨГДӨЛ БЭЛТГЭХ НЬ**  
**(Preparation of open NUM handwritten character dataset)**

Мэдээллийн технологи (D061303)  
Бакалаврын судалгааны ажил

Улаанбаатар

2020 он

**МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ**  
**ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААН, ИНЖЕНЕРЧЛЭЛИЙН СУРГУУЛЬ**  
**МЭДЭЭЛЭЛ, КОМПЬЮТЕРИЙН УХААНЫ ТЭНХИМ**

**ГАР БИЧМЭЛ ТАНИЛТАНД ЗОРИУЛСАН МУИС-Н**  
**НЭЭЛТТЭЙ ӨГӨГДӨЛ БЭЛТГЭХ НЬ**  
**(Preparation of open NUM handwritten character dataset)**

Мэдээллийн технологи (D061303)  
Бакалаврын судалгааны ажил

Удирдагч: \_\_\_\_\_ Др. Б.Сувдаа

Гүйцэтгэсэн: \_\_\_\_\_ Ц.Хасбилэгт (15B1SEAS0980)

Улаанбаатар

2020 он

# Зохиогчийн баталгаа

Миний бие Цэрэнбямбаагийн Хасбилэгт "ТАР БИЧМЭЛ ТАНИЛТАНД ЗОРИУЛСАН МУИС-Н НЭЭЛТТЭЙ ӨГӨГДӨЛ БЭЛТГЭХ НЬ" сэдэвтэй судалгааны ажлыг гүйцэтгэсэн болохыг зарлаж дараах зүйлсийг баталж байна:

- Ажил нь бүхэлдээ эсвэл ихэнхдээ Монгол Улсын Их Сургуулийн зэрэг горилохоор дэвшүүлсэн болно.
- Энэ ажлын аль нэг хэсгийг эсвэл бүхлээр нь ямар нэг их, дээд сургуулийн зэрэг горилохоор оруулж байгаагүй.
- Бусдын хийсэн ажлаас хуулбарлаагүй, ашигласан бол ишлэл, зүүлт хийсэн.
- Ажлыг би өөрөө (хамтарч) хийсэн ба миний хийсэн ажил, үзүүлсэн дэмжлэгийг дипломын ажилд тодорхой тусгасан.
- Ажилд тусалсан бүх эх сурвалжид талархаж байна.

Гарын үсэг: \_\_\_\_\_

Огноо: \_\_\_\_\_

## Contents

ЗУРГИЙН ЖАГСААЛТ .....	iii
ХҮСНЭГТИЙН ЖАГСААЛТ .....	iv
УДИРТГАЛ .....	1
Зорилго .....	2
Зорилт .....	2
БҮЛГҮҮД .....	3
1. СУДАЛГАА .....	3
1.1 Нийтлэг ашиглагддаг сангууд .....	3
1.2 Судалгаанаас хийсэн дүгнэлт .....	7
2. АРГЫН ТОДОРХОЙЛОЛТ .....	8
2.1 Оролт .....	8
2.2 Боловсруулалт .....	8
2.3 Гаралт .....	8
3. ХӨГЖҮҮЛЭЛТ .....	9
4. ДҮГНЭЛТ .....	10
НОМ ЗҮЙ .....	11
ХАВСРАЛТ .....	12
A. СЭРЭЭХ ҮГ ИЛРҮҮЛЭГЧ .....	12
B. KALDI GSTREAMER СЕРВЕРТ ЗОРИУЛСАН ТОХИРГООНЫ ФАЙЛ .....	13
C. ЯРИА ТАНИХ ХЭСЭГ .....	14
D. ЧАДВАР СОНГОХ .....	15
E. ЦАГ, ОГНООНЫ МЭДЭЭЛЭЛ АВАХ .....	17
F. ЯРИА ҮҮСГҮҮР .....	18

## List of Figures

Зураг	Хуудас
1.1 NIST SD19 сангийн гар бичмэл тэмдэгтүүд цуглуулах зорилготойгоор бэлдсэн маягт. Гараар бөглөх 33 хэсгүүдтэй ба нэр, огноо оруулах нүднээс бусад хэсгүүд боловсруулагдана. ....	4
1.2 MNIST сангийн жишээ тэмдэгтүүд [3] .....	5
1.3 EMNIST санг үүсгэхэд NIST сангийн зургуудыг боловсруулахад ашигласан хувиргалтын дараалал [2] .....	6

## List of Tables

Хүснэгт

Хуудас

## **УДИРТГАЛ**

Өнөөдөр хиймэл оюун ухаан болон машин сургалтын салбаруудын технологи, судалгаанууд хөгжихийн хэрээр, өмнө нь хүний төсөөлж байгаагүй олон шинэ боломжууд, зөвхөн технологийн салбаруудаар хязгаарлагдалгүй, ар араасаа нээгдэж байгаагаас гадна өнөөдөр нийтэд ашиглагдаж буй технологиудын нарийвчлал, үр дүн, чанар ч мөн үүнийг дагаад сайжирч байгаа билээ. Энэ дундаас гар бичмэл танилттай холбоотой судалгаанууд олноор хөгжүүлэгдэх нь бидний мэдэх хүн компьютерийн харилцааг эергээр өөрчилж байна.

## **Зорилго**

Энэ чиглэлийн судалгаанууд, түүн дээр суурилсан програм хангамж, үйлчилгээ, бүтээгдэхүүн хөгжүүлэгдэхэд гар бичмэлийг таних, машин сургалтанд ашиглах өгөгдөл зайлшгүй хэрэгтэй, мөн Монгол хэл дээр энэ төрлийн нээлттэй, нэгдсэн өгөгдлийн сан хомс учир энэхүү судалгааны ажлаар өөрийн эх хэл дээрх нээлттэй гар бичмэлийн санг бий болгох, санг үүсгэхэд ашиглах арга, хэрэгслийг тодорхойлохыг зорьсон юм.

## **Зорилт**

Монгол хэл дээрх гар бичмэлийн нээлттэй санг үүсгэхдээ дараах үе шатын дагуу ажиллана.

1. Үндсэн шаардлагуудыг тодорхойлох
2. Тэмдэгтүүд бичүүлж авах маягтын загварыг гаргах
3. Бөглөгдсөн маягт дээр боловсруулалт хийх, тэмдэгтүүдийг ялгаж авах
4. Их хэмжээний оролтын үед боловсруулалт хийх боломжтой байдлаар програмыг хөгжүүлэх
5. Програмыг PyPi, GitHub дээр байршуулах, програмтай холбоотой мэдээлэл, ашиглах зааврыг бичих



# 1. СУДАЛГАА

Машин сургалт тэр дундаа гар бичмэл танилттай холбоотой машин сургалтын өгөгдлийг хэрхэн бэлтгэх, зохион байгуулах тал дээрх бодитой, албан ёсны судалгааны ажлууд хомс байсан тул өгөгдлийн хэлбэр, хэмжээ, санг хэрхэн зохион байгуулах зэргийг энэ чиглэлийн өөр бусад хэл дээрх түгээмэл ашиглагддаг нээлттэй өгөгдлүүдийг судалсаны үндсэн дээр тэдгээрийн жишигт тааруулан зохион байгуулахаар шийдсэн.

Англи, хятад, япон, солонгос хэл дээрх хамгийн нийтлэг ашиглагддаг сангуудыг судалж үзээд анх өгөгдлийг хэрхэн цуглуулсан болон яаж боловсруулалт хийгдсэн талаарх мэдээлэл хэр хангалттай байснаас харгалзан үзээд дараах англи хэл дээрх дөрвөн сангуудын хэрхэн боловсруулагдсан байдлыг дэлгэрэнгүй судлалаа.

## 1.1 Нийтлэг ашиглагддаг сангууд

### 1.1.1 *NIST 19 special database (SD19)*

1995 оны 3 дугаар сард NIST буюу National Institute of Standards and Technology газраас танилцуулагдсан энэ сан нь нийт 3699 хуудас бүхий хар, цагаан гараар бөглөгдсөн маягт (Зураг 1.1), тэдгээрээс ялгаж авсан 814255 тоо, үсгүүдийн цуглуулгаас бүрдэж байсан. Зөвхөн тоо ([0-9]), үсэг ([a-z][A-Z]) оролцсон учир нийт 62 төрлийн тэмдэгтүүдийг, тэмдэгт бүр нь 128x128 харьцаатайгаар зохион байгуулагдсан. [1]

Цуглуусан өгөгдлөө `by_write`, `by_field`, `by_class`, `by_merge` хэмээх дөрвөн өөр төрлөөр ангилсан бөгөөд өгөгдлүүдээ хадгалахдаа дараах файлын төрлүүдээр хадгалсан. Үүнд:

- `.mis` ялган авсан тэмдэгтийн зургуудыг хадгалах файлын төрөл
- `.pct` бүтэн хуудас маягтын зургийг хадгалах файлын төрөл
- `.cls` 62 ялгаатай төрлийн тэмдэгтүүдийн холбогдох зургийн мэдээллийг хамт хадгалж буй файлын төрөл

# HANDWRITING SAMPLE FORM

NAME [REDACTED] DATE 8-3-89 CITY MINNEN CITY STATE MI. ZIP 48452

This sample of handwriting is being collected for use in testing computer recognition of hand printed numbers and letters. Please print the following characters in the boxes that appear below.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9

87 701 3752 80759 960941

158 4586 32123 832656 82

7481 80539 419219 67 904

61738 729658 75 390 5716

109334 40 625 4234 46002

gyxlakpdsbtisurumwfqjenhocv

9YXKqKp4567890123456789

ZXSBNGECEMYWQTKFLUOHPIRVDJA

ZXSBNGECEMYWQTKFLUOHPIRVDJA

Please print the following text in the box below:

We, the People of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common Defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our posterity, do ordain and establish this CONSTITUTION for the United States of America.

We, the People of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common Defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our posterity, do ordain and establish this CONSTITUTION for the United States of America.

Figure 1.1: NIST SD19 сангийн гар бичмэл тэмдэгтүүд цуглуулах зорилготойгоор бэлдсэн маягт. Гараар бөглөх 33 хэсгүүдтэй ба нэр, огноо оруулах нүднээс бусад хэсгүүд боловсруулагдана.

- .mit маяг бөглөсөн хүний мэдээллийг холбогдох тэмдэгтүүдийн мэдээлэлтэй хамт хадгалж буй файлын төрөл

Эхний хувилбараасаа хойш 2016 оны 9 сард санг хадгалах файлын төрлийг PNG форматтай болгон шинэчилж дахин шинэ хувилбар гаргажээ.

## 1.1.2 MNIST [3]

Анх 1998 онд дээр дурдсан NIST -н Special Database 1 болон 3 дахь хувилбаруудын 0 -с 9 хүртэл 10 төрлийн тэмдэгтүүдтэйгээр танилцуулагдсан бөгөөд өнөөдрийг хүртэл оюун ухаан, хиймэл оюуны салбарт хамгийн түгээмэл ашиглагддаг сан яах аргагүй мөн билээ.

NIST анхлан SD-3 нь сургалтанд, NIST SD-1 нь тестэд зориулан зохион байгуулсан

бөгөөд учир нь сургалтанд зориулсан өгөгдлүүд нь нөгөөхөөсөө хамаагүй цэвэрхэн, сайн боловсруулагдсан, амар танигдахаар байсан бол SD-1 нь ахлах сургуулийн сурагчдаар бөглүүлсэн маягтаас боловсруулагдсан байсан юм. Тийм ч учраас эдгээрх сангуудыг тэнцүү хуваан, дахин зохион байгуулж сургалтын 60000, тестийн 10000 зурагтай MNIST (Modified NIST) -г үүсгэсэн байна.[4]



Figure 1.2: MNIST сангийн жишээ тэмдэгтүүд [3]

NIST -н оролтын хар, цагаан зургуудыг эхлээд 20x20 хэмжээтэйгээр багасган боловсруулж үндсэн гурван хэлбэрээр туршсан. Эхнийх нь хэмжээг нь багасгасан зургуудынхаа төвийг олон, харьцааг алдагдуулахгүйгээр 28x28 дотор багтаасан. Хоёр дахь хувилбарт оролтын зургийн налалтыг арилган тэгшлээд, 20x20 хэмжээтэй болгон багасгаж deslanted буюу налууг арилгасан санд ашигласан. Харин сүүийнх нь хэлбэр нь хэмжээг нь 16x16 болгон багасгаж ашигласан.

Өнөөдөр MNIST сан нь дээрх төрлүүдийн хамгийн эхний хэлбэрээр вектор, олон хэмжээст матриц хадгалахад амар IDX гэх файлын төрөлтэйгөөр ашиглагдаж байгаа билээ.

### 1.1.3 EMNIST [2]

Extended Modified NIST буюу EMNIST нь аль хэдийн MNIST санг ашиглаж байгаа алгоритм, системүүд ямар ч асуудалгүйгээр авч ашиглах боломжтой байх илүү өргөтгөсөн нээлттэй санг үүсгэх зорилготойгоор танилцуулагдсан юм.

Оролтын зургийг боловсруулах дараалал нь ерөнхийдөө MNIST[3] -г үүсгэх зорилгоор NIST[1] -н өгөгдлүүдийг хэрхэн боловсруулсан аргатай ойролцоо ч зургийг жижигрүүлэхэд ашигласан алгоритм, арга зэрэг нь өөр байгаа юм. Зурган дээр хувиргалт хийхдээ эхлээд зургийн урт, өргөнөөс хамгийн их хэмжээгээр хамгийн жижиг багтаасан тэгш өнцөгтийг тооцоолж хоосон үлдсэн хэсэгт дүүргэлт хийснээр урт, өргөнийг тэнцүүлэн дөрвөлжин зураг үүсгэсэн. Зургийн хэсгүүдийг захын ирмэгүүдэд наалдуулахгүйн тулд мөн 2 пикселийн хүрээг нэмжээ. Эндээс зургаа bi-cubic interpolation алгоритмт ашиглан 28x28 хэмжээтэй болгон жижигрүүлсэн байна. Зураг 1.3.

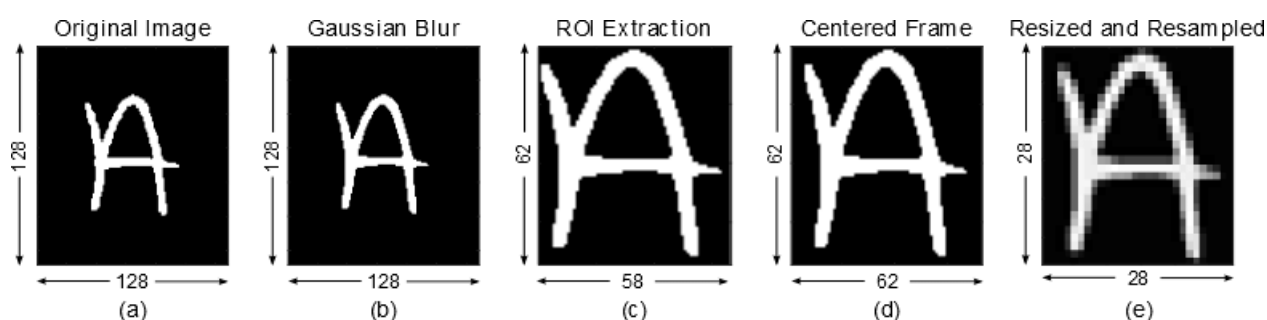


Figure 1.3: EMNIST санг үүсгэхэд NIST сангийн зургуудыг боловсруулахад ашигласан хувиргалтын дараалал [2]

Машин сургалтанд зориулагдсан ямар ч сан сургалтын болон тестийн гэж хуваагддаг бөгөөд EMNIST -н хувьд NIST -н өгөгдлийг боловсруулахдаа MNIST -н баримталсан аргыг буюу Census Bureau -н ажилчид, ахлах сургуулийн сурагчдаас цуглуулсан тэмдэгтүүдийг сургалт болон тестийн өгөгдөлдөө тэнцүү хуваан, тестийн өгөгдөл үүсгэхдээ тэдгээрээс санамсаргүй аргаар сонгон авч үүсгэжээ.

### 1.1.4 IAM [6]

Англи хэл дээрх хамгийн түгээмэл ашиглагддаг сангуудын нэг нь IAM<sup>1</sup> сан бөгөөд дээр дурдсан сангуудаас ялгаатай нь үндсэн өгөгдөл нь дан гараар бичигдсэн бүтэн өгүүлбэрээс цуглуулагдсан буюу LOB corpus [5]<sup>2</sup> дээр суурилж энэхүү сан боловсруулагдсан. LOB corpus -н анхны хувилбар 1970 онд танилцуулагдсан бөгөөд 400 гаруй хүний 1066 гараар бөглөгдсөн маягтаас авсан 10841 ялгаатай үгсийг агуулдаг.

## 1.2 Судалгаанаас хийсэн дүгнэлт

Өнөөдөр машин сургалт, гар бичмэл танилтын судалгаануудад хамгийн нийтлэг ашиглагддаг англи хэлний дөрвөн сангуудад судлагаа хийж үзээд дараах дүгнэлтүүдийг гаргалаа. Үүнд:

- Сан нь хэрхэн зохион байгуулагдсан, өгөгдлүүд ямар форматтайгаар хадгалагдсан, авч ашиглах, хувиргах боломжууд нь энэ чиглэлээр судалгаа хийж буй хүмүүсийн дунд тархах, түгээмэл ашиглагдахтай шууд холбоотой. Иймээс санг авч ашиглах, өгөгдлүүдийг хувиргахад амар байх
- MNIST болон EMNIST нь энэ чиглэлийн сургалт, судалгаа, шинэ төслүүдэд хамгийн түгээмэл ашиглагдаж буй сан мөн бөгөөд үүнтэй холбоотойгоор эдгээр сангуудыг ашигладаг олон систем, програм хангамжууд байгаа учир өөрийн үүсгэж буй сангаа эдгээр сангуудтай нийцтэй ажиллах боломжтой байдлаар зохиомжлох
- Сангийн сургалтын болон тестийн өгөгдлүүдийг хэрхэн ялгаж, зохион байгуулах нь чухал бөгөөд гаргацтай зургууд, танихад хэцүү зургуудыг аль аль талд тэнцүү байдлаар хуваах

---

<sup>1</sup>Institut für Informatik und angewandte Mathematik (= Компьютерийн ухаан, хэрэглээний математикийн тэнхим), Берны их сургууль, Берн, Швейцарь

<sup>2</sup>Lancaster-Oslo болон Bergen сургуулиудын хамтран боловсруулсан сан <http://korpus.uib.no/ibase/manuals/LOB/INDEX.HTM>

## **2. АРГЫН ТОДОРХОЙЛОЛТ**

### **2.1 Оролт**

### **2.2 Боловсруулалт**

### **2.3 Гаралт**

### **3. ХӨГЖҮҮЛЭЛТ**

## 4. ДҮГНЭЛТ

Энэхүү судалгааны ажлаар бусад хэл дээрх гар бичмэл танилттай холбоотой судалгаануудын ажлууд дээр тулгуурлан өөрийн хэлний гар бичмэл танилтанд зориулсан нээлттэй сан, түүнд ашиглах програм хангамжийг хөгжүүлэхээр зорьлоо. Монгол хэл дээрх гар бичмэл танилт нь оюун ухаан, машин сургалтын салбарын энгийн судалгааны ажил биш бөгөөд бичиг баримтыг дижиталжуулах, хүний бичгийн хэвийг ялган таних (authentication), баталгаажуулах (authorization) гэх мэт гар бичмэлтэй холбоотой асар олон төрлийн ажил, судалгаа, програм хангамжуудад ч мөн ашиглагдах боломжтой. Тийм ч учраас дээрх салбаруудтай холбоотой ажил, судалгаа хийхээр зорьж буй хүмүүст энэхүү судалгааны ажил суурь судалгаа нь болох бөгөөд хамгийн чухал нь Монгол хэлний гар бичмэлтэй холбоотой судалгаануудад ашиглагдах нэгдсэн нэг өгөгдлийн сан болон цаашид тогтмол хөгжүүлэгдэж байх хэрэгтэй.

Нийт өгөгдлийн тоо хэмжээ, төрлүүдийг нэмэх, өргөжүүлэхийн тулд python хэл дээрх програм хангамжийн боломжуудыг нэмэх, бичвэр агуулах маягтыг динамик байдлаар үүсгэх, ажиллагааг хурдасгах, нийтээс маягтын зургаа авч автоматаар боловсруулан сангаа шинэчлэх бие даасан вэбсайт, систем гэх мэт ажлууд нэмэгдэж хийх боломжтой.



# Ашигласан материал

- [1] NIST Special Database 19  
<https://www.nist.gov/srd/nist-special-database-19>
- [2] Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017).  
*EMNIST: an extension of MNIST to handwritten letters.*  
<http://arxiv.org/abs/1702.05373>
- [3] MNIST — Wikipedia, The Free Encyclopedia  
[https://en.wikipedia.org/wiki/MNIST\\_database](https://en.wikipedia.org/wiki/MNIST_database)
- [4] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner *Gradient-based learning applied to document recognition* in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.
- [5] S. Johansson, G.N. Leech, H. Goodluck: *Manual of information to accompany the Lancaster-Oslo/Bergen corpus of British English, for use with digital computers.* Department of English, University of Oslo, Oslo, (1978)
- [6] Marti, U., Bunke, H. *The IAM-database: an English sentence database for offline handwriting recognition.* IJDAR 5, 39–46 (2002). <https://doi.org/10.1007/s100320200071>
- [7] ETLCD — ETL Character Database  
<http://etlcdb.db.aist.go.jp/obtaining-etl-character-database>

# A. СЭРЭЭХ ҮГ ИЛРҮҮЛЭГЧ

```
1 from precise_runner import PreciseEngine, PreciseRunner
2 import time
3 from pixel_ring import pixel_ring # ReSpeaker led controller
4 import mraa
5 import os
6
7 # (pixel ring)
8 en = mraa.Gpio(12)
9 if os.getuid() != 0:
10     time.sleep(1)
11 en.dir(mraa.DIR_OUT)
12 en.write(0)
13 pixel_ring.set_brightness(20)
14
15 # \ \ \ \
16 def on_activation():
17     print('hello')
18     pixel_ring.wakeup() # \ \ \ \
19     # ...
20     # \ ,\ \ \ \ \ \ \ \ ...
21     # ...
22     time.sleep(3)
23     pixel_ring.off()
24
25 def main():
26     # \ \ \
27     engine = PreciseEngine('precise-engine/precise-engine', 'hey-obi.
28         pb') # hey-obi.pb\ \ \
29     runner = PreciseRunner(engine, on_activation=on_activation)
30     runner.start()
31
32 if __name__ == '__main__':
33     main()
34     while True:
35         time.sleep(1)
```

# В. KALDI GSTREAMER СЕРВЕРТ ЗОРИУЛС ТОХИРГООНЫ ФАЙЛ

```
1 timeout-decoder : 10
2 decoder:
3     model: /opt/models/tri3b/final.mdl
4     lda-mat: /opt/models/tri3b/final.mat
5     word-syms: /opt/models/tri3b/graph/words.txt
6     fst: /opt/models/tri3b/graph/HCLG.fst
7     silence-phones: "1:2:3:4:5"
8     left-context: 3
9     right-context: 3
10 out-dir: tmp
11
12 use-vad: False
13 silence-timeout: 5
14
15 ...
```

# С. ЯРИА ТАНИХ ХЭСЭГ

```
1 def stt():
2     audio_file = '.tmp/stt.flac'
3     text = False
4     # \ \
5     subprocess.run(['rec', '-c', '1', '-r', '16000', '-d', audio_file
6                     ,
7                     'trim', '0', '15', 'silence', '1', '0.1', '0.3%',
8                     '1', '3.0', '0.3%'])
9     # Curl\ \ \ HTTP \
10    output = str(subprocess.check_output(
11        "curl -T {} {}".format(audio_file, conf.config()['STT_URL']),
12        shell=True), 'utf-8')
13    print(output)
14    if output == 'No workers available':
15        return False
16    response = json.loads(output)
17    print(response)
18    # \ \ \ \ \ \ \ \ \ \
19    if response['status'] == 0:
20        text = response['hypotheses'][0]['utterance']
21    return text
```

## D. ЧАДВАР СОНГОХ

```
1 result = {'obj': None, 'answer': ''}
2 if intent['intent']['confidence'] > 0.5:
3     # intent \ \
4     #
5     if intent['intent']['name'] == 'greeting':
6         result['answer'] = greeting.greet()
7     #
8     elif intent['intent']['name'] == 'get_time':
9         result['answer'] = time_n_date.get_time()
10    #
11    elif intent['intent']['name'] == 'get_date':
12        result['answer'] = time_n_date.get_date()
13    # \ \ \ \
14    elif intent['intent']['name'] == 'get_day':
15        result['answer'] = time_n_date.get_day()
16    # \
17    elif intent['intent']['name'] == 'get_weather':
18        p = {}
19        # entity \ \ \ \ \ \ \ \
20        for entity in intent['entities']:
21            p[entity['entity']] = entity['value']
22        result['answer'] = weather.get_weather(**p)
23    # \
24    elif intent['intent']['name'] == 'get_currency':
25        if len(intent['entities']) > 0:
26            result['answer'] = currency.get_currency(
27                intent['entities'][0]['value'])
28        else:
29            result['answer'] = currency.get_currency()
30    #
31    elif intent['intent']['name'] == 'order':
32        if len(intent['entities']) > 0:
33            result['obj'] = order.Order(intent['entities'][0]['value'])
34        else:
35            result['obj'] = order.Order()
36            result['answer'] = ' {} '.format(
37                result['obj']._name)
38    elif intent['intent']['name'] == 'order_size':
39        if len(intent['entities']) > 0:
40            result['obj'] = obj.set_size(intent['entities'][0]['value'])
41        else:
42            result['obj'] = obj.set_size()
43            result['answer'] = ' '
44    elif intent['intent']['name'] == 'order_quantity':
45        if len(intent['entities']) > 0:
46            obj.set_quantity(intent['entities'][0]['value'])
47        else:
48            obj.set_quantity()
49            result['answer'] = obj.order()
50    else:
```

```
51     result = False
52 #     \     \     \     \     \     \     \
53 if result == False:
54     result = {'answer': DO_NOT_KNOW[math.floor(
55         random.random() * len(DO_NOT_KNOW))], 'obj': obj}
56 return result
```

## Е. ЦАГ, ОГНООНЫ МЭДЭЭЛЭЛ АВАХ

```
1 from datetime import date, datetime
2
3 WEEKDAY_NAMES = [' ', ' ', ' ', ' ', ' ', ' ', ' ']
4
5 def get_time():
6     now = datetime.now()
7     hour = now.hour
8     minute = now.minute
9     return str(hour) + ' ' + str(minute) + ' '
10
11 def get_day():
12     today = date.today()
13     day = WEEKDAY_NAMES[today.weekday()]
14     return ' ' + day + ' '
15
16 def get_date():
17     today = date.today()
18     month = today.month
19     day = today.day
20     return ' ' + str(month) + ' ' + str(day) + ' - '
```

# Б. ЯРИА ҮҮСГҮҮР

```
1 # \ \ \ \
2 def get_wav(sentence=''):
3     url = config()['TTS_URL'] + str(sentence)
4     r = requests.get(url)
5     with open('.tmp/tts.wav', 'wb') as f:
6         f.write(r.content)
7     return './.tmp/tts.wav'
8
9 # \ \
10 def play_wav(location='.tmp/tts.wav'):
11     chunk = 1024
12     f = wave.open(location, "rb")
13     p = pyaudio.PyAudio()
14     stream = p.open(format=p.get_format_from_width(f.getsampwidth()),
15                     channels=f.getnchannels(), rate=f.getframerate(), output=True
16                     )
17     data = f.readframes(chunk)
18     while data:
19         stream.write(data)
20         data = f.readframes(chunk)
21     stream.stop_stream()
22     stream.close()
23     p.terminate()
```