Mobile AI XPlore: Research Project Term Final Report

Yuqing Yang

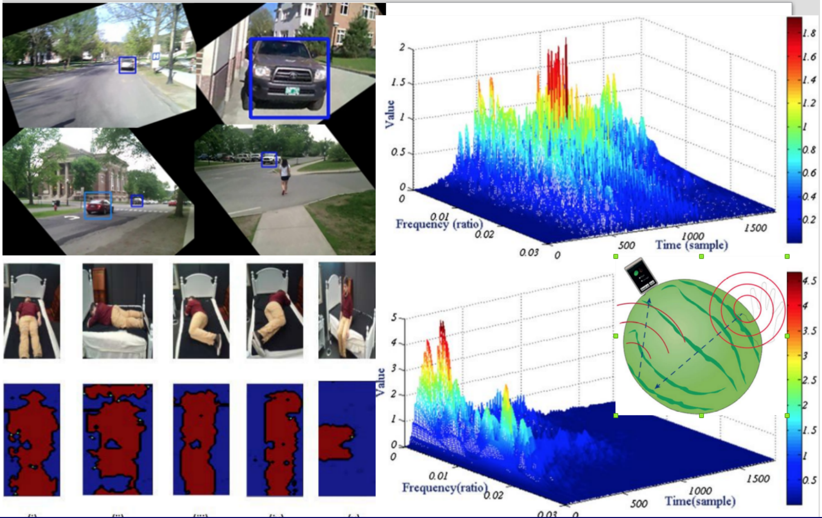
Nanjing University

December 28, 2018

Outline

1. Introduction
   - Mobile AI
   - Potential Problems
   - Background & Related Works
2. Challanges
   - Decompilation
   - Model Encryption
3. Experiment Procedures
   - Target
   - Exploration Setup
4. Findings and Conclusions
   - Mobile AI Workflow
   - Conclusion

Mobile AI


Value
Frequency (ratio)
Time (sample)
0.01
0.02
0.03
500
1000
1500
0
0.5
1
1.5
2
1.8
1.6
1.4
1.2
0.8
0.6
0.4
0.2
Value
Frequency(ratio)
Time(sample)
5
4
3
2
1
4.5
3.5
2.5
1.5
0.5

There are huge numbers of AI models deployed on mobile apps:

- Classification & Inferring
- Recognition & Detection
- Translation
- Prediction
- ...

Mobile AI

There are numerous frameworks for mobile AI deployment:

- tensorflow mobile
- tensorflow lite
- caffe
- caffe2
- ncnn(tencent)
- ...

Potential problems on mobile AI

Large usage of models shed potential problems on both users and companies

- Users:
  - Information Leaking
  - Personal Privacy
- Companies:
  - Model stealing
  - Framework Crashes

ncnn

- Mobile AI framework from Tencent
- Provides model translation service
- Can be deployed on mobile apps
- Models can be crypted

Drawbacks on Traditional Fuzzing

- Mainly based on flows
    - Hard to adapt
- Mainly based on Models
    - Requies source files

Outline

1. Introduction
   - Mobile AI
   - Potential Problems
   - Background & Related Works
2. Challanges
   - Decompilation
   - Model Encryption
3. Experiment Procedures
   - Target
   - Exploration Setup
4. Findings and Conclusions
   - Mobile AI Workflow
   - Conclusion

Decompilation

- Symbol Elimination
- Code Irreversibility

Model Encryption

- Convert souce param file into bin
- Data Unavailability
- Input Schema Unavailability

Outline

1 Introduction
- Mobile AI
- Potential Problems
- Background & Related Works

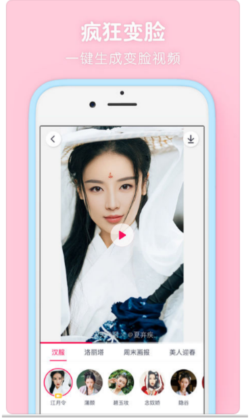
2 Challanges
- Decompilation
- Model Encryption

3 Experiment Procedures
- Target
- Exploration Setup

4 Findings and Conclusions
- Mobile AI Workflow
- Conclusion

Target

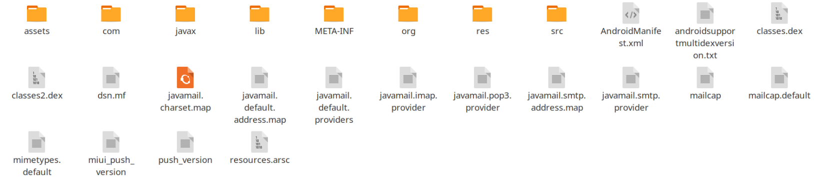
- TianTianPTu(Tencent)



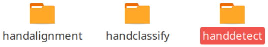
Exploration Setup

Apk

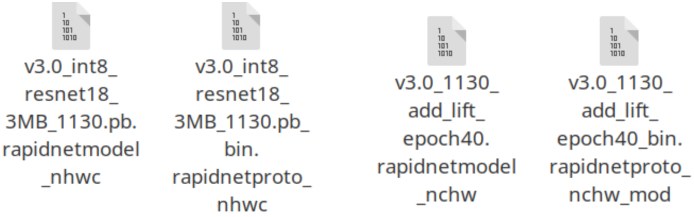
Extract content:



assets
com
javax
lib
META-INF
org
res
src
AndroidManifest.xml
androidsupportmultidexversion.txt
classes.dex
classes2.dex
dsn.mf
javamail.charset.map
javamail.default.address.map
javamail.default.providers
javamail.imap.provider
javamail.pop3.provider
javamail.smtp.address.map
javamail.smtp.provider
mailcap
mailcap.default
mimetypes.default
miui_push_version
push_version
resources.arsc



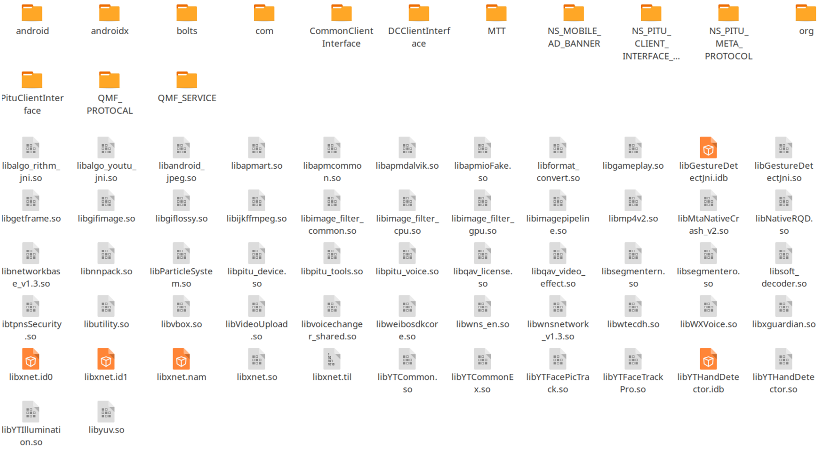
handalignment
handclassify
handdetect



v3.0_int8_resnet18_3MB_1130.pb.rapidnetmodel_nhwc
v3.0_int8_resnet18_3MB_1130.pb_bin.rapidnetproto_nhwc
v3.0_1130_add_lift_epoch40.rapidnetmodel_nchw
v3.0_1130_add_lift_epoch40_bin.rapidnetproto_nchw_mod

Exploration Setup

Extract Libraries:



android
androidx
bolts
com
CommonClientInterface
DCClientInterface
MTT
NS_MOBILE_AD_BANNER
NS_PITU_CLIENT_INTERFACE_...
NS_PITU_META_PROTOCOL
org
PituClientInterface
QMF_PROTOCAL
QMF_SERVICE

libalgo_rithm_jni.so
libalgo_youtu_jni.so
libandroid_jpeg.so
libapmart.so
libapmcommon.so
libapmdalvik.so
libapmioFake.so
libformat_convert.so
libgameplay.so
libGestureDetectJni.idb
libGestureDetectJni.so
libgetframe.so
libgifimage.so
libgiflossy.so
libijkffmpeg.so
libimage_filter_common.so
libimage_filter_cpu.so
libimage_filter_gpu.so
libimagepipeline.so
libmp4v2.so
libMtaNativeCrash_v2.so
libNativeRQD.so
libnetworkbase_v1.3.so
libnnpack.so
libParticleSystem.so
libpitu_device.so
libpitu_tools.so
libpitu_voice.so
libqav_license.so
libqav_video_effect.so
libsegmentern.so
libsegmentero.so
libsoft_decoder.so
ibtpnsSecurity.so
libutility.so
libvbox.so
libVideoUpload.so
libvoicechanger_shared.so
libweibosdkcore.so
libwns_en.so
libwnsnetwork_v1.3.so
libwtecdh.so
libWXVoice.so
libxguardian.so
libxnet.id0
libxnet.id1
libxnet.nam
libxnet.so
libxnet.til
libYTCommon.so
libYTCommonEx.so
libYTFacePicTrack.so
libYTFaceTrackPro.so
libYTHandDetector.idb
libYTHandDetector.so
libYTIllumination.so
libyuv.so

Analyze the workflow

- A thourough analysis into 7,222 java files
- 105,9909+ LoC
- 57 C++ libraries
- 2 different deployed models

Outline

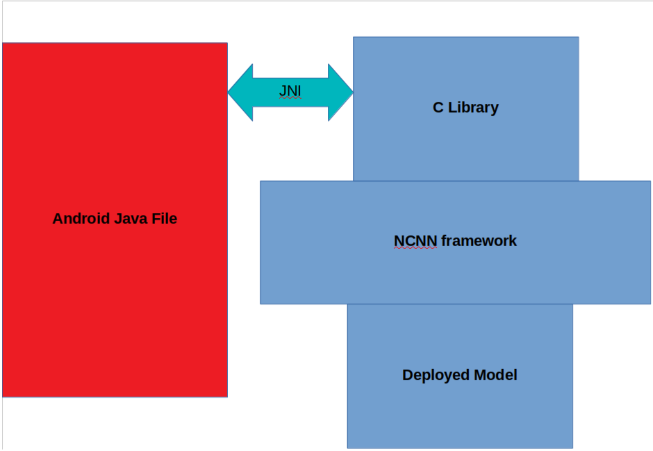
1 Introduction
- Mobile AI
- Potential Problems
- Background & Related Works

2 Challanges
- Decompilation
- Model Encryption

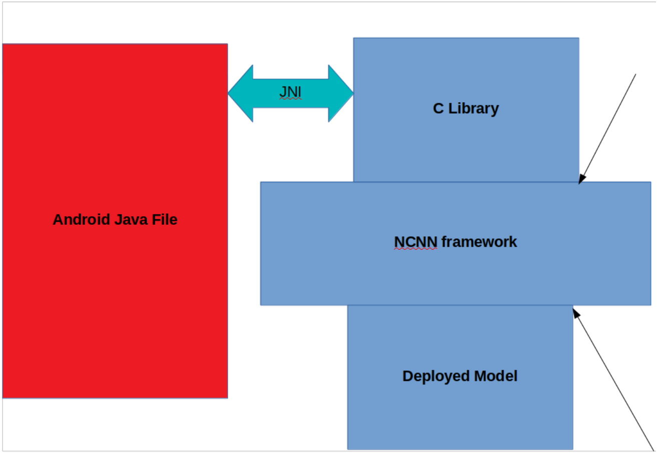
3 Experiment Procedures
- Target
- Exploration Setup

4 Findings and Conclusions
- Mobile AI Workflow
- Conclusion

Mobile AI Workflow



Potential Injection Point


JNI
C Library
Android Java File
NCNN framework
Deployed Model